

# Probabilistic Robust Autoencoders for Outlier Detection

Ofir Lindenbaum <sup>1\*</sup> Yariv Aizenbud <sup>2\*</sup>

Yuval Kluger<sup>1†</sup>

<sup>1</sup> Bar-Ilan University, Israel; <sup>2</sup>Yale University, USA;

<sup>†</sup>Corresponding author. E-mail: [yuval.kluger@yale.edu](mailto:yuval.kluger@yale.edu)

Address: 333 Cedar St, New Haven, CT 06510, USA

\* These authors contributed equally.

## Abstract

Anomalies (or outliers) are prevalent in real-world empirical observations and potentially mask important underlying structures. Accurate identification of anomalous samples is crucial for the success of downstream data analysis tasks. To automatically identify anomalies, we propose Probabilistic Robust AutoEncoder (PRAE). PRAE aims to simultaneously remove outliers and identify a low-dimensional representation for the inlier samples. We first present the Robust AutoEncoder (RAE) objective as a minimization problem for splitting the data into inliers and outliers. Our objective is designed to exclude outliers while including a subset of samples (inliers) that can be effectively reconstructed using an AutoEncoder (AE). RAE minimizes the autoencoder’s reconstruction error while incorporating as many samples as possible. This could be formulated via regularization by subtracting an  $\ell_0$  norm counting the number of selected samples from the reconstruction term. Unfortunately, this leads to an intractable combinatorial problem. Therefore, we propose two probabilistic relaxations of RAE, which are differentiable and alleviate the need for a combinatorial search. We prove that the solution to the PRAE problem is equivalent to the solution of RAE. We use synthetic data to show that PRAE can accurately remove outliers in a wide range of contamination levels. Finally, we demonstrate that using PRAE for anomaly detection leads to state-of-the-art results on various benchmark datasets.

Unsupervised anomaly detection is a fundamental problem in data mining and machine learning. The goal is to identify unusual measurements in unlabeled datasets. Identifying anomalous samples is essential for empirical science in various fields, such as biology [1], geophysics [2], engineering, and cyber-security [3]. Anomalies and outliers are samples that significantly deviate from the “normal” (often majority) observations. A critical challenge is defining such normality and automatically identifying all abnormal measurements.

One solution for anomaly detection relies on the data’s density. By estimating the data density, anomalies could be identified as samples drawn from the low probability density regions [4]. Density-based models include Local Outlier Factor (LOF) [5], or some of its variants [6, 7, 8]. More recent probabilistic approaches include [9, 10]. Other schemes [11, 12, 13, 14] rely on distances between samples to identify anomalies; the basic assumption is that normal points have dense neighborhoods while outliers are far from their neighbors. Another paradigm for anomaly detection is one-class-classification, where anomalies are identified as samples that significantly deviate from the major part of the data (i.e. the “one-class”) [15, 16, 17, 18].

High dimensional measurements may often be described by a low dimensional subspace or manifold [19, 20, 21]. By assuming that the normal samples lie near a low dimensional latent

manifold, while outliers are diverse and do not follow the same manifold structure, the anomalies could be detected via a dimensionality reduction method, such as Principal Component Analysis (PCA) [22], or deep Autoencoders (AE) [23, 24, 25]. Robust PCA schemes [26] seek for a low dimensional linear subspace that “best” fits the inliers. These models can identify anomalies and learn a reduced sup-space simultaneously; however, they are restricted to linear transformations. To overcome this limitation, several authors have proposed to use AEs [27, 28] to learn a valuable nonlinear mapping for detecting outliers.

Generative models are powerful tools for learning the data distribution and, therefore, turn valid for anomaly detection [29, 30, 31, 32]. In the more relaxed semi-supervised setting, several authors [33, 34, 35, 36, 37] have used deep neural networks to model the normal part of the data; and identify outliers as samples that deviate significantly from normal samples. Recently, there has been growing interest in anomaly detection in vision, self-supervision [38, 39], and transfer learning [40] can be used to learn feature maps that capture the normal part of the data.

In this work, we focus on unsupervised anomaly detection for general data (not necessarily images) and propose a novel Probabilistic Robust autoencoder (PRAE). We note that in this work we treat anomalies and outliers in the same way since we only assume that they deviate from the “normal” samples. PRAE can simultaneously remove outliers and learn a low-dimensional representation of the inlier samples. Our contributions are four folds: (1) We formulate the robust autoencoding (robust-AE) problem by incorporating an  $\ell_0$  term penalizing the number of observations included in the AE’s reconstruction loss. (2) We propose two probabilistic relaxations for robust AE and demonstrate that they could be effectively trained using standard optimization tools such as stochastic gradient descent. (3) We show theoretically that the solution of the probabilistic relaxation is equivalent to the solution to the robust-AE problem. (4) We demonstrate using extensive simulations that PRAE outperforms leading anomaly detection methods in multiple settings.

# 1 Background

**Notation:** Throughout the paper we denote vectors using bold lowercase letters such as  $\mathbf{x}$ . Scalars are denoted by lower case letters such as  $y$ . The  $n^{th}$  vector-valued observation is denoted as  $\mathbf{x}_n$  while  $x[d]$  represents the  $d^{th}$  feature (or entry) of the vector  $\mathbf{x}$ . Matrices are denoted by bold uppercase letters  $\mathbf{X}$ . The  $\ell_p$  norm of  $\mathbf{x}$  is denoted by  $\|\mathbf{x}\|_p$ .

## 1.1 Autoencoder (AE)

The AE is a multilayer neural network designed for dimensionality reduction. It comprises an encoder and decoder, which are typically symmetric and are trained jointly to minimize the reconstruction error of the data. The number of neurons in the latent space (the hidden layer between the encoder and decoder) controls the dimension of the reduced representation of the data. It has been shown that the reduced representation of a linear AE spans the same subspace as the principal components of the data [41]. Although it has been suggested to use an AE without dimensionality reduction [42], in this work all AE have low dimensional latent space.

Given samples  $X = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$ , where  $\mathbf{x}_i \in \mathbb{R}^D$ , the AE learns the reduced representation by minimizing the following reconstruction loss

$$\frac{1}{N} \sum_i \left\| \mathbf{x}_i - \hat{\mathbf{x}}_i \right\|_2^2, \quad (1)$$

where the reconstructed vector  $\hat{\mathbf{x}}_i$  is obtained as the output of the encoder  $\rho()$  and decoder  $\psi()$ , i.e.  $\hat{\mathbf{x}}_i = \psi(\rho(\mathbf{x}_i))$ . The encoder-decoder pair are defined using a multi-layer neural network; this can be described using the following equations:

$$\begin{aligned}\rho_\ell(\mathbf{x}) &= \sigma\left(\mathbf{W}_{\ell-1}^\rho \rho_{\ell-1}(\mathbf{x}) + \mathbf{b}_{\ell-1}^\rho\right), \text{ and} \\ \psi_\ell(\mathbf{z}) &= \sigma\left(\mathbf{W}_{\ell-1}^\psi \psi_{\ell-1}(\mathbf{z}) + \mathbf{b}_{\ell-1}^\psi\right), \ell = 1, \dots, L,\end{aligned}$$

where  $\rho^{(0)}(\mathbf{x}) = \mathbf{x}$  and  $\psi_0(\mathbf{z}) = \mathbf{z} = \rho_L(\mathbf{x})$ . The weights  $\mathbf{W}_\ell^\rho$ ,  $\mathbf{W}_\ell^\psi$  and biases  $\mathbf{b}_\ell^\rho$ ,  $\mathbf{b}_\ell^\psi$  at each layer  $\ell$  are learnt by applying stochastic gradient descent to the reconstruction loss. The operator  $\sigma$  is a nonlinear activation function applied in an element-wise fashion.

## 2 Method

### 2.1 Robust Autoencoder

Given samples  $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$ , where  $\mathbf{x}_i \in \mathbb{R}^D$ , we model the data by  $\mathbf{X} = \mathbf{X}_{in} \cup \mathbf{X}_{out}$ , where  $\mathbf{X}_{in}$  are inliers and  $\mathbf{X}_{out}$  are outliers. We assume that  $\mathbf{X}_{in}$  can be approximated by some low dimensional structure. Our goal is to identify the inliers and outliers. We propose to use a regularized AE that simultaneously learns a low dimensional representation of the data and identifies the outliers. We define an indicator vector  $\mathbf{b} \in \{0, 1\}^N$  whose value  $i$  indicates if the sample  $\mathbf{x}_i$  is an inlier ( $b[i] = 1$ ) or an outlier ( $b[i] = 0$ ). To learn the parameters of the encoder-decoder pair ( $\rho()$  and  $\psi()$ ) while simultaneously identifying the inliers and outliers, we propose the following robust AE loss

$$L_d(\psi, \rho, \mathbf{b}) = \sum_i b[i] \left\| \mathbf{x}_i - \hat{\mathbf{x}}_i \right\|_2^2 - \lambda \|\mathbf{b}\|_0, \quad (2)$$

where  $\hat{\mathbf{x}}_i = \psi(\rho(\mathbf{x}_i))$ . The leading term in Eq. (2) is a standard AE reconstruction term computed only for samples with  $b[i] = 1$ . The  $\ell_0$  norm in Eq. (2) counts the number of samples that are included in the reconstruction error; these samples are tagged as ‘‘inliers’’. By balancing the reconstruction error and the  $\ell_0$  penalty, the hyper-parameter  $\lambda$  controls the cost associated with the number of samples used by the AE. A large value of  $\lambda$  will force the model to include more samples. On the other hand, a small  $\lambda$  would lead to a sparser solution with fewer samples included by the model. If  $\mathbf{X}_{in}$  lie near a low dimensional manifold, we assume that the encoder-decoder pair can lead to a good approximation of the inliers, that is  $\hat{\mathbf{x}}_i \approx \mathbf{x}_i$  for  $\mathbf{x}_i \in \mathbf{X}_{in}$ . On the other hand, if outliers do not lie near the low dimensional manifold, we expect  $\|\mathbf{x}_i - \hat{\mathbf{x}}_i\|_2^2$  to be large. Unfortunately, due to the  $\ell_0$  norm in Eq. (2) the problem becomes intractable even for a small number of samples. To overcome this limitation, following [43, 44, 45], we propose to replace the deterministic search over the values of the indicator vector  $\mathbf{b}$  with a probabilistic counterpart.

### 2.2 Probabilistic Autoencoder

We are now ready to present our probabilistic formulation for a sparse AE. Towards this goal, we multiply the samples by stochastic gates that relax the binary nature of the indicator vector  $\mathbf{b}$ . The gates are differentiable and are purposed to select a subset of samples on which the AE reconstruction error is minimized. We parameterize a stochastic gate (STG) using mean shifted

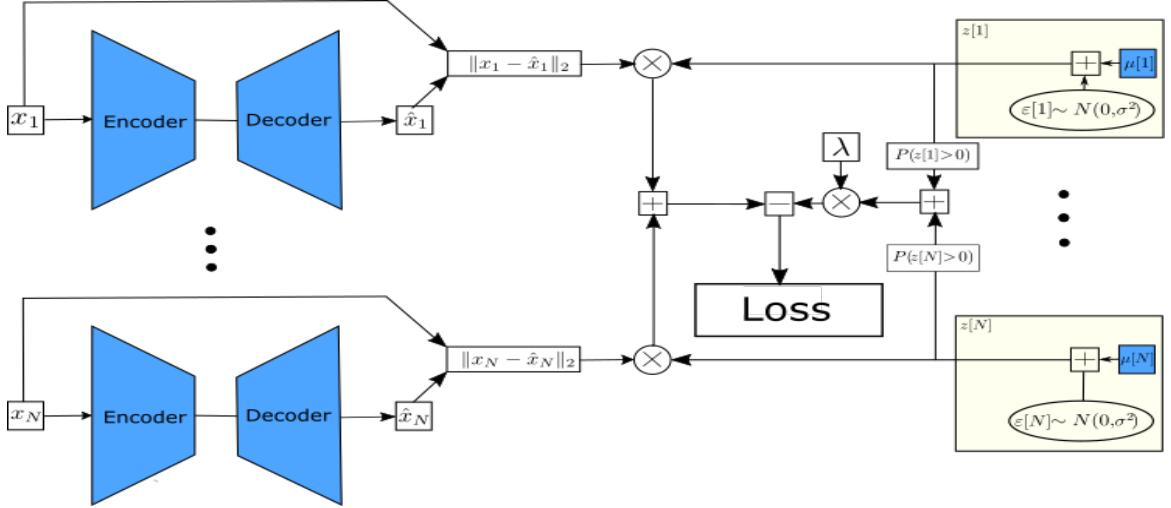


Figure 1: A schematic of the PRAE- $\ell_0$  algorithm (see Eq. (4)). The  $\mathbf{x}_i$ s (on the left) are the input samples. For any choice of encoder, decoder, and  $\boldsymbol{\mu}$  (all in blue), a reconstruction loss is computed (middle) with a subtracted term designed to encourage the model to select only the inlier samples (see Eq. 4). We optimize over the “blue” variables (one encoder, one decoder, and a vector  $\boldsymbol{\mu}$  that result in the lowest loss). The resulting value  $\mu[i]$  serves as the outlier score for data point  $\mathbf{x}_i$ .

truncated Gaussian distribution. Specifically, we denote the STG random vector as  $\mathbf{z} \in [0, 1]^N$ , parametrized by  $\boldsymbol{\mu} \in \mathbb{R}^N$ . Each vector entry is defined as

$$z[i] = \max(0, \min(1, \mu[i] + \epsilon[i])), \quad (3)$$

where  $\epsilon[i]$  is drawn from  $\mathcal{N}(0, \sigma^2)$ ,  $\sigma$  is fixed throughout training, and  $\mu[i]$  is a trainable parameter which controls the distribution of the random variable  $z[i]$ .

We can now incorporate the STGs into our proposed probabilistic AE loss. Formally, using the reconstruction loss of (1), a probabilistic AE loss can be described using one of the following terms

$$L_{p_0}(\boldsymbol{\psi}, \boldsymbol{\rho}, \boldsymbol{\mu}) = \mathbb{E}\left(\sum_i z[i] \|\mathbf{x}_i - \hat{\mathbf{x}}_i\|_2^2 - \lambda \|z\|_0\right), \quad (4)$$

$$L_{p_1}(\boldsymbol{\psi}, \boldsymbol{\rho}, \boldsymbol{\mu}) = \mathbb{E}\left(\sum_i z[i] \|\mathbf{x}_i - \hat{\mathbf{x}}_i\|_2^2 - \lambda \|z\|_1\right), \quad (5)$$

where,  $\lambda$  is a regularization parameter that controls the cost associated with the number of samples included by the AE. To minimize the new loss functions (4) or (5), we propose the following strategy. Given some initial guess for the encoder, decoder, parameterized via the weights of  $\boldsymbol{\psi}$  and  $\boldsymbol{\rho}$ , we draw realizations for the random vector  $\mathbf{z}$  and compute the loss value. We note that the regularization terms  $\mathbb{E}(\|z\|_0) = \sum P(z[i] > 0)$  and  $\mathbb{E}(\|z\|_1) = \sum \mathbb{E}(z[i])$  are parametric, and the expected value of the left term of Eqs. (4) and (5) is approximated using Monte Carlo sampling. Then, we differentiate the loss using Stochastic Gradient Descent to update the weights in  $\boldsymbol{\psi}$  and  $\boldsymbol{\rho}$ , and the vector  $\boldsymbol{\mu}$ . A schematic illustration of this procedure is presented in Figure 1. After convergence, we use the values of the trainable vector  $\boldsymbol{\mu}$  as anomaly scores for each data point. A smaller value indicates that the sample should be excluded from the reconstruction loss and therefore is more anomalous. In Section 5.2 we propose two unsupervised schemes for tuning the regularization parameter  $\lambda$ .

### 3 Related Work

The problem of anomaly detection has been previously addressed using AEs. A straightforward solution is to train a standard AE and use the reconstruction error of each sample to quantify if it is normal or anomalous [46]. Since this approach does not include regularization, the AE may overfit the outliers and learn a mapping that does not correctly characterize the normal samples. To solve this limitation, in [27] the authors propose an ensemble of AEs for anomaly detection. The idea is to train many AEs, each pruned by randomly subsampling the learned connectivities. Then, an aggregated prediction of the ensemble is used to identify the anomalies. One disadvantage of this approach is that it requires extensive computational and memory costs since it involves training hundreds of AEs. Furthermore, the proposed scheme outperforms the ensemble of AEs on several benchmark datasets (see results in supplemental Section S3).

Perhaps the most related method to our work is [28]. In [28], the authors proposed the following regularized AE objective

$$\|\mathbf{L}_D - \psi(\rho(\mathbf{L}_D))\|_2 + \lambda \|\mathbf{S}\|_{2,1} \quad s.t. \quad \mathbf{X} - \mathbf{L}_D - \mathbf{S} = 0.$$

This model aims to split the data  $\mathbf{X}$  into two parts,  $\mathbf{L}_D$  and  $\mathbf{S}$  while minimizing the reconstruction error on  $\mathbf{L}_D$ . The regularization in the form of an  $\ell_{2,1}$  norm is designed to sparsify the rows (samples) or columns (features) of  $\mathbf{S}$ . This way, the data is split into inliers  $\mathbf{L}_D$  and a sparse set of outliers  $\mathbf{S}$ . To minimize the objective with the additional constraint, they use the Alternating Direction Method of Multipliers (ADMM) [47] with an element-wise projection approach to enforce the constraint. This method differs from our approach significantly since the regularization relies on the  $\ell_{2,1}$  norm applied to  $\mathbf{S}$ . The  $\ell_{2,1}$  norm leads to shrinkage of values, and therefore  $\mathbf{S}$  is not guaranteed to reflect actual samples from  $\mathbf{X}$ . Furthermore, it relies on a different optimization scheme and is not amenable to parallelization through small batch training (due to the additional constraint). Our model overcomes these limitations and, as demonstrated in section 5 leads to more accurate identification of outliers when applied to real and synthetic data.

### 4 Analysis

In this section, we justify the use of our proposed probabilistic AE (see Section 2.2) to solve the robust auto encoding problem (see Section 2.1). Since the latter is not differentiable while the first is, our goal is to show that both minimization problems lead to the same solution.

First, to avoid divergence of the values of  $\mu$  in the theoretical analysis, we bound the values of  $\mu$  by

$$-M \leq \mu[i] \leq M, \tag{6}$$

for some large number  $M$ . Note that the number  $M$  is used only for the theoretical analysis and has no practical use when running the algorithm.

For any vector  $\mathbf{b} \in \{0, 1\}^N$ , we define  $\mu_b$ , such that  $\mu_b[i] = -M$  if  $\mathbf{b}[i] = 0$ , and  $\mu_b[i] = M$  if  $\mathbf{b}[i] = 1$  for  $i = 1 \dots N$ . For any  $\mu$  we define  $\mathbf{b}_\mu$  such that  $\mathbf{b}_\mu[i] = \text{sign}(\mu[i])$ .

We now turn our attention to show that the deterministic optimization problem (2) (which is not differentiable) is equivalent to our probabilistic optimization (5) in the following sense

**Theorem 4.1.** *For any dataset  $\mathbf{X}$ , denote by  $(\psi_d, \rho_d, \mathbf{b}_d)$  the minimizer of (2) and by  $(\psi_p, \rho_p, \mu_p)$  the minimizer of (5). Assume that the minimizer of (2) is unique and that*

$$\min_{(\psi, \rho, \mathbf{b}) \neq (\psi_d, \rho_d, \mathbf{b}_d)} L_d(\psi, \rho, \mathbf{b}) \geq L_d(\psi_d, \rho_d, \mathbf{b}_d) + \varepsilon_0 \tag{7}$$

for some  $\varepsilon_0 > 0$ . Then for a sufficiently large  $M > 0$  (see (6)),  $(\psi_d, \rho_d) = (\psi_p, \rho_p)$ , and for any  $i = 1, \dots, L$ ,  $b[i] = 1$  if  $\mu[i] > 0$  and  $b[i] = 0$  otherwise.

In other words if the minimizer of (2) is unique then, the encoder, decoder that minimize (2) and (5) are equivalent. Moreover, the samples included by both models (indicated by  $\mathbf{b}$  and  $\mathbf{z}$ ) are the same.

*Proof.* The proof construction is comprised of three arguments. The final argument relies on the first two and concludes the proof.

**Argument 1:** For any triplet  $(\psi_d, \rho_d, \mathbf{b})$  the deterministic loss  $L_d$  can be approximated by the probabilistic loss  $L_{p_1}$ . Namely, for any  $\varepsilon, \delta > 0$  there is a value of  $M > 0$  such that

$$|L_d(\psi_d, \rho_d, \mathbf{b}_d) - L_{p_1}(\psi_d, \rho_d, \boldsymbol{\mu}_b)| \leq \varepsilon,$$

with probability  $1 - \delta$ .

To prove this argument, we first compute the expectation  $E(z)$  using definition (3), we get:

$$\begin{aligned} E(z) = & \mu - \frac{1}{\sqrt{2\pi}} \int_{-\infty}^0 t e^{-\frac{(t-\mu)^2}{2\sigma^2}} dt - \frac{1}{\sqrt{2\pi}} \int_1^\infty t e^{-\frac{(t-\mu)^2}{2\sigma^2}} dt \\ & + \frac{1}{\sqrt{2\pi}} \int_1^\infty e^{-\frac{(t-\mu)^2}{2\sigma^2}} dt, \end{aligned}$$

computing the integrals leads to:

$$\begin{aligned} E(z) = & \frac{\sigma}{\sqrt{2\pi}} (e^{-\frac{\mu^2}{2\sigma^2}} - e^{-\frac{(1-\mu)^2}{2\sigma^2}}) + (\mu - 1) * \Phi\left(\frac{1-\mu}{\sigma}\right) \\ & - \mu * \Phi\left(-\frac{\mu}{\sigma}\right) + 1, \end{aligned}$$

where  $\Phi$  is the CDF of the standard normal distribution.

Since  $\lim_{\mu \rightarrow \infty} E(z) = 1$ , and  $\lim_{\mu \rightarrow -\infty} E(z) = 0$ , than, for any  $\varepsilon > 0$ , there is a sufficiently large  $M$ , such that

$$\left| \lambda \sum_i \mathbb{E}(z[i]) - \lambda \|\mathbf{b}\|_0 \right| < \varepsilon/2. \quad (8)$$

From the definition of  $z$  we also know that for  $\mu > 1$ ,  $P(z \neq 1) = \Phi\left(\frac{1-\mu}{\sigma}\right)$ , and thus  $\lim_{\mu \rightarrow \infty} P(z = 1) = 1$ . Similarly  $\lim_{\mu \rightarrow -\infty} P(z = 0) = 1$ . Thus, for any  $\delta$  there is  $M$  large enough, such that

$$\left| \sum_i z[i] \|\mathbf{x}_i - \hat{\mathbf{x}}_i\|_2^2 - \sum_i b[i] \|\mathbf{x}_i - \hat{\mathbf{x}}_i\|_2^2 \right| < \varepsilon/2, \quad (9)$$

with probability  $1 - \delta$ .

Combining (8) and (9), we have that for any  $\delta > 0$ , there is a value of  $M$  such that

$$|L_d(\psi_d, \rho_d, \mathbf{b}_d) - L_{p_1}(\psi_d, \rho_d, \boldsymbol{\mu}_b)| \leq \varepsilon,$$

with probability  $1 - \delta$ . This concludes the proof of Argument 1.

**Argument 2:** For any AE  $(\psi, \rho)$ , the minimum  $\min_{\boldsymbol{\mu}} L_{p_1}(\psi, \rho, \boldsymbol{\mu})$  is achieved when  $\boldsymbol{\mu}[i]$  equals to either  $M$  or  $-M$  for all  $i$ .

Assume by contradiction that the minimum of  $L_{p_1}$  is achieved at a point where for some  $k$ ,  $\boldsymbol{\mu}[k]$  is not either  $M$  or  $-M$ . If  $\|\mathbf{x}_i - \hat{\mathbf{x}}_i\|_2^2 \geq \lambda$ , then for  $\hat{\boldsymbol{\mu}}$  such that  $\hat{\boldsymbol{\mu}}[i] = \boldsymbol{\mu}[i]$  for all  $i \neq k$

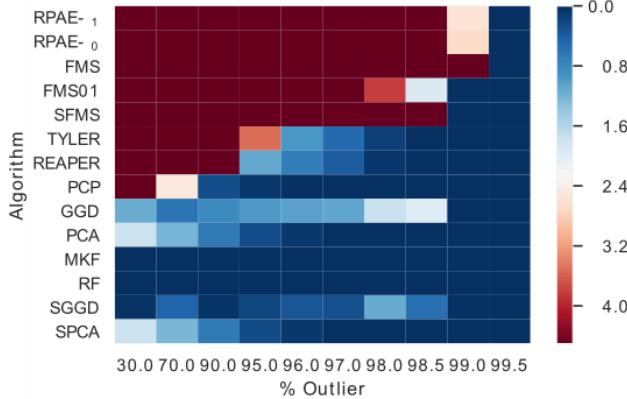


Figure 2: Comparing PRAE to several Robust Subspace Recovery (RSR) algorithms. The y-axis represents the different algorithms, and the x-axis represents different percentiles of outliers. Each box is colored according to the mean over 10 runs of the log of the angle between the recovered subspace and the ground truth.

and  $\hat{\mu}[k] = -M$  we have that  $L_{p_1}(\psi, \rho, \hat{\mu}) \leq L_{p_1}(\psi, \rho, \mu)$ , which contradicts the minimality of  $L_{p_1}(\psi, \rho, \mu)$ . In case  $\|\mathbf{x}_i - \hat{\mathbf{x}}_i\|_2^2 \leq \lambda$  a similar argument will lead to a contradiction as well.

**Argument 3:** Assume by contradiction that the minimizers of (2) and (5) are not equivalent, i.e.

$$(\psi_d, \rho_d, \mu_{b_d}) \neq (\psi_p, \rho_p, \mu_p). \quad (10)$$

From Argument 2 we have that  $\mu_p[i] = M$  or  $-M$  for all  $i$ . From Argument 1 we have that

$$\begin{aligned} \|L_d(\psi_p, \rho_p, b_{\mu_p}) - L_{p_1}(\psi_p, \rho_p, \mu_p)\| &\leq \varepsilon \text{ and} \\ \|L_d(\psi_d, \rho_d, b_d) - L_{p_1}(\psi_d, \rho_d, \mu_{b_d})\| &\leq \varepsilon. \end{aligned} \quad (11)$$

Since  $\psi_p, \rho_p, \mu_p$  is the minimizer of  $L_{p_1}$ , we have from (11) that

$$L_d(\psi_d, \rho_d, b_d) \geq L_d(\psi_p, \rho_p, b_{\mu_p}) - 2\varepsilon. \quad (12)$$

From Eq. (10) and the assumption of the theorem in Eq. (7), we have that

$$\|L_d(\psi_p, \rho_p, b_{\mu_p}) - L_d(\psi_d, \rho_d, b_d)\| \geq \varepsilon_0. \quad (13)$$

For  $2\varepsilon < \varepsilon_0$ , Eq. (13) contradicts Eq. (12).  $\square$

## 5 Experiments

### 5.1 Robust Subspace Recovery Problem

First, we test the performance of the proposed algorithms in the linear setting. While this regime has fewer applications, it is well studied, and it is easier to analyze and compare different methods.

We note that in the linear regime, the outlier detection problem is strongly related to the Robust Subspace Recovery problem (RSR). Thus, this section focuses on comparing our proposed scheme to baseline methods designed to solve the RSR problem. The RSR problem involves finding a low-dimensional (linear) subspace in a corrupted, potentially high-dimensional dataset. For a complete overview of RSR, we refer the reader to [26].

Following [26], for any chosen percentile of outliers  $r = N_{out}/N$ , we generate  $N = 10000$  points in  $\mathbb{R}^{200}$  in the following way: first we randomly generate  $\mathbf{X}_{in}^{low}$ , a set of  $N_{in} = (1 - r)N$  random points in  $\mathbb{R}^{10}$ . Next we generate a random linear transformation  $\mathbf{T} \in \mathbb{R}^{200 \times 10}$ , and set  $\mathbf{X}_{in}^{high} = \mathbf{T}\mathbf{X}_{in}^{low}$ . Finally, we generate  $\mathbf{X}_{out}$  as  $N_{out} = rN$  random points in  $\mathbb{R}^{200}$ , and define the dataset  $\mathbf{X} = \mathbf{X}_{in}^{high} \cup \mathbf{X}_{out}$ . The task is to recover  $\mathbf{T}$  and  $\mathbf{X}_{in}$  given the data  $\mathbf{X}$ . The accuracy is measured by the log of the angle between the recovered  $\mathbf{T}$  and the correct  $\mathbf{T}$ . Each experiment was performed 10 times, and the final outcome is the average of the 10 runs. For complete implementation details we refer the reader to supplemental Section S1.

The result of the comparison to other algorithms under different percentile of outliers ( $r = N_{out}/N$ ) appears in Figure 2. We compared to leading schemes from the evaluation in [26], namely: fast median subspace (FMS) [48], Tyler’s M-estimator (TYLER) [49], REAPER [50], the augmented Lagrange multiplier method (PCP)[51], geodesic gradient descent (GGD) [52], and principal component analysis (PCA).

While our approach is not explicitly designed for the RSR problem, it is easy to see that our algorithms perform on par with state-of-the-art methods for RSR. Even for 99% outliers, in 7 out of 10 runs, our algorithms found exactly all the inliers. We note that since our approach is not designed for RSR and focused on the more general non-linear setting, FMS recovers a more accurate subspace and requires a shorter training time. Nonetheless, we argue that this experiment highlights that our model is relatively robust to the number of outliers. We observe that PRAE can correctly recover inliers in a noisy setting. Precisely, when we use  $\mathbf{X}_n = \mathbf{X} + \eta$ , where the noise  $\eta \sim N(0, 10^{-2}I)$  and 99% outliers, our model is still able to correctly identify a subset of inliers that are sufficient for subspace recovery in 7 out of 10 cases. However, we omit this noisy RSR experiment since it requires a more involved method for reconstructing the subspace-based on the (correctly) retrieved noisy inliers.

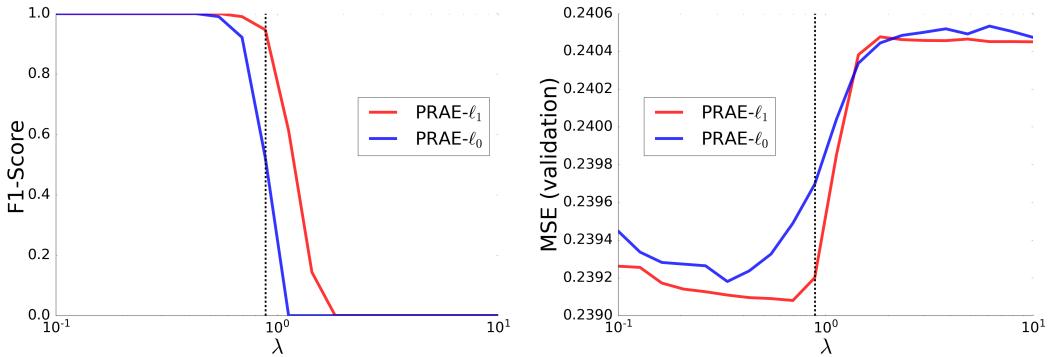


Figure 3: Phase transition of PRAE. As we increase  $\lambda$  above a certain threshold, PRAE starts to include outliers resulting in a lower F1-score (left panel) and larger reconstruction error (MSE) on unseen samples (right panel). Dashed line indicates our (unsupervised) estimation of the value of  $\lambda$  in which the proposed scheme transitions from removal to inclusion of outliers.

## 5.2 Unsupervised Schemes for Tuning the Regularization Parameter

One practical consideration in PRAE is the choice of regularization parameter  $\lambda$ . In this section, we empirically study the effect of this parameter and propose two unsupervised schemes for tuning  $\lambda$ . Then, we use synthetic data to demonstrate that our estimated value of  $\lambda$  leads to accurate identification of all inliers and removal of all outliers.

We focus on the linear data model described in section 5.1, but with  $N = 200$ ,  $\mathbf{X}_{in}^{low} \in \mathbb{R}^{150 \times 2}$ , and  $\mathbf{T} \in \mathbb{R}^{100 \times 2}$ . We generate data from this model and run PRAE- $\ell_0$ , and PRAE- $\ell_1$  for various

values of  $\lambda$  in the range  $[0.1, 10]$ . We run each model 20 time and record the average F1-score which is computed based on precision and recall of outlier identification. Specifically, we define an outlier  $\mathbf{x}_i$  as a sample such that after training  $\mu[i] < \text{thresh}$ , and an inlier otherwise. Here,  $\mu[i]$  is computed based on Eq. 3 but without the injected Gaussian noise. We set  $\text{thresh}$  to 0.1 although other values within  $(0, 1)$  lead to similar results.

In both proposed loss functions (see (4) and (5))  $\lambda$  balances between the number of sample included by the model and the reconstruction loss. For a very large  $\lambda$ , we expect the model to include all samples since the regularization term would be larger than the reconstruction of  $\mathbf{x}_i$  (for inliers and outliers). On the other hand, if  $\lambda = 0$  all samples should be excluded by the model. For small values of  $\lambda > 0$ , we expect the model to include the inliers (since we can obtain zero reconstruction loss) and exclude the outliers. Based on the linear model experiment (described above), we observe a "phase transition" in the behavior of PRAE as a function of  $\lambda$ . Namely, as evident in the left panel of Figure 3 for small values of  $\lambda$ , PRAE accurately removes all outliers and includes all inliers.

In this example, since all samples have roughly the same energy ( $\ell_2$  norm), we can propose a simple scheme for estimating the  $\lambda$  value in which the phase transition occurs. Specifically, we can compute the mean energy of all samples, namely  $ME = \frac{1}{N} \sum_i^N \|\mathbf{x}_i\|_2^2$ . Since we can not reliably reconstruct the outliers (based on our data model), we expect the error for reconstructing outliers to be  $\sim ME$ . Therefore, for any  $\lambda > ME$ , PRAE- $\ell_1$  should include outliers since  $\|\mathbf{x}_i - \hat{\mathbf{x}}_i\|_2^2$  is compared to  $\lambda$  in loss (see (5)). On the other hand, if  $\lambda < ME$ , PRAE- $\ell_1$  should exclude outliers (based on the same argument). For PRAE- $\ell_0$ , this argument is not precise; nonetheless, we observe that ME lines well with the phase transition of both models. This is presented as a dashed black line in Figure 3.

Another scheme to tune the regularization parameter is based on the value of  $\lambda$  that minimizes the reconstruction loss of unseen samples (a validation set). Here, we assume that inliers can be represented by a low dimensional space while outliers can not. By evaluating the model's reconstruction error on unseen samples, we can check if the model suffered from overfitting on anomalies or has used only inliers. We repeat the experiment above but evaluate the average reconstruction loss on 200 unseen samples generated from the same model. As evident in the right panel of Figure 3 both models lead to the smallest reconstruction loss for  $\lambda$  values that coincide with the perfect F1-score (supporting the validity of the proposed tuning scheme). We observe that PRAE- $\ell_0$  leads to a higher reconstruction error for large values of  $\lambda$ . This might indicate that the inclusion of all samples occurs earlier in training leading to stronger overfitting.

### 5.3 Evaluation Based on Synthetic Data

To illustrate the qualities of our scheme in a nonlinear setting, we use a simple artificial example suggested by [53]. For the normal samples, we consider a "narrow swiss-roll", with 1000 points uniformly sampled from  $[3\pi/2, 9\pi/2] \times [0, 0.1]$ , and embedded into  $\mathbb{R}^3$  using  $(t, h) \rightarrow (t \cos(t), h, t \sin(t))$ . Then, we generate additional 200 "outliers" sampled from  $N(0, \sigma_N^2 I_3)$ , where  $I_3 \in \mathbb{R}^{3 \times 3}$  is the identity matrix. We generate such data with several values of  $\sigma_N^2$ , and following [27, 54] we evaluate the quality of multiple anomaly detection methods using Receiver Operating Characteristic (ROC) curves. The ROC measures the trade-off between true positive and false positive rates. The true positive rate is defined as the ratio between identified anomalies and true anomalies, while the false positive rate is the portion of normal samples identified as anomalies. The ROC curves are summarized by measuring the AUC (area under the curve). We compare our method to several strong baselines with code available at [55]. Specifically, we use CBLOF [56] a local clustering-based approach, COF [57] which uses the density of the

data, SOD [58] and HBOS [59] which are based on proximity, IForest [60] and LSCP [61] which are ensemble methods. We also compare to the probabilistic ABOD [62] and to OC-SVM [63]. Finally, we compare to several NN based methods, such as:  $\ell_{2,1} - AE$  [28], Deep-SVDD [16], and RSR-AE [53]. Table 1 shows the median AUC of all baselines over ten runs. For all NN models (including PRAE), we use five hidden layers and a latent dimension of 2. Since the inliers lie near a manifold (“swiss-roll”) with an intrinsic dimension of 2, the AE can correctly capture the swiss-roll’s mapping. As indicated by table 1, PRAE accurately distinguishes inliers from outliers for several scales of  $\sigma_N^2$ .

$\sigma_N^2$	CBLOF	ABOD	COF	IForest	SOD	LSCP	HBOS	OC-SVM	DSVDD	RSR-AE	$\ell_{2,1}-AE$	PRAE- $\ell_0$	PRAE- $\ell_1$
0.1	13.81	48.77	63.39	26.08	31.54	67.72	60.65	68.54	69.19	93.12	68.47	<b>96.23</b>	<b>97.48</b>
1	28.88	54.67	64.64	55.20	76.35	67.99	81.34	50.58	71.97	82.25	46.23	<b>99.97</b>	<b>99.94</b>
10	93.77	53.01	98.55	97.87	93.28	99.65	94.34	52.81	40.75	75.13	80.59	<b>99.55</b>	<b>99.42</b>

Table 1: Performance comparison on a synthetic dataset. Each value corresponds to the median AUC of the ROC curve of a different algorithm (column) under different anomaly variance (row). Blue and bold indicate first and second ranked methods respectively.

## 5.4 Anomaly Detection on Real Data

Next, we evaluate the proposed approach on a diverse set of real-world datasets whose properties appear in the right column of table 2. Since we focus on unsupervised anomaly detection, we wish to identify anomalies in two settings: (1) “in-sample”, in which we aim to curate a given dataset from outliers, and (2) “out-of-sample”, in which we want to identify outliers from newly arrived samples without additional training. To evaluate performance on (1), we apply our algorithm to the entire dataset and use the value of the gate parameter  $\mu[i]$  as an anomaly score for  $x_i$ . For (2), we split each dataset and use one half for training the PRAE while the hold-out-set is used to evaluate the performance without additional training. Here, since we don’t have gates from unseen samples, the anomaly score for  $x_i$  is based on the reconstruction error  $\|x_i - \hat{x}_i\|_2^2$ .

dataset	CBLOF	ABOD	COF	IForest	SOD	LSCP	HBOS	OC-SVM	DSVDD	RSR-AE	$\ell_{2,1}-AE$	PRAE- $\ell_0$	PRAE- $\ell_1$	N/D / Out %
Lympho	95.12	72.53	88.08	<b>99.98</b>	92.08	97.38	<b>99.01</b>	97.88	81.13	63.90	98.72	96.09	95.98	148/ 18/ 4.1
Ecoli	87.90	75.72	87.77	86.41	82.67	86.71	81.44	86.71	92.07	76.80	44.21	<b>88.94</b>	<b>89.09</b>	336/ 7/ 2.7
Cardio	85.31	49.87	51.79	92.15	69.19	59.42	88.14	58.39	61.89	69.44	88.10	<b>94.28</b>	<b>93.87</b>	1831/ 21/ 9.6
Yeast	68.63	56.81	53.18	79.30	61.79	61.62	78.98	62.16	66.62	79.54	70.58	<b>83.37</b>	<b>83.95</b>	1364/ 8/ 4.8
Musk	95.17	68.04	52.31	97.46	89.93	60.45	98.91	90.28	80.17	82.59	<b>98.96</b>	<b>99.17</b>	98.61	3062/ 166/ 3.1
Thyroid	91.68	48.73	58.34	<b>97.64</b>	87.57	81.91	<b>95.06</b>	84.38	90.09	90.12	81.39	94.78	94.69	3772/ 6/ 2.1
MNIST-S	52.97	67.67	69.93	83.37	70.64	92.18	77.09	90.20	85.01	91.29	90.39	<b>93.54</b>	<b>93.31</b>	5127/ 784/ 5.2
Fashion-MNIST	89.04	50.65	69.25	91.79	64.74	82.96	74.75	88.87	78.28	82.86	90.69	<b>93.89</b>	<b>93.99</b>	5300/ 784/ 5.6
PBMC	76.36	51.50	90.71	88.71	79.48	NA	70.36	88.69	55.67	51.10	64.26	90.91	<b>91.30</b>	6300/ 32738/ 4.7
Pendigits	91.73	63.13	52.33	95.18	64.91	47.39	92.69	88.14	78.82	89.54	93.73	<b>97.56</b>	<b>97.47</b>	6870/ 16/ 2.2
Mammog.	81.42	83.00	71.18	86.62	77.51	50.53	83.00	81.29	83.76	82.40	81.31	<b>88.31</b>	<b>88.33</b>	11183/ 6/ 2.3
Shuttle	66.50	58.22	53.59	<b>99.67</b>	50.74	53.29	98.42	63.66	97.89	82.31	97.78	<b>99.13</b>	98.95	49097/ 9/ 7.1
Credit	NA	NA	NA	95.01	NA	NA	94.36	<b>95.39</b>	58.78	84.57	92.86	<b>95.25</b>	91.31	284807/ 16/ 0.17
Average-AUC	83.21	62.51	67.71	91.14	76.41	70.05	86.15	84.36	76.02	78.67	82.93	<b>93.01</b>	<b>93.46</b>	
Median rank	6	12	12	3	9	10	2	7	9	7	5	2	2	

Table 2: Performance comparison with several leading baselines. We present the median AUC over 10 runs. Blue and bold indicate first and second-ranked methods respectively. NA indicates that the baseline did not converge on the dataset.

We train our proposed AE with an encoder-decoder pair with five hidden layers each of size 10; the hidden dimension is 1 (this might not be optimal but worked well across most datasets). We use the heuristic proposed in section 5.2 to tune the regularization parameter to  $\lambda = 1 < ME$ . We run all methods 10 times and record the ROC for each run. In table 2 we

present the median AUC of the proposed method and all baselines evaluated for the “in-sample” setting. These results show that the proposed approach compares favorably to leading methods on a wide range of datasets. Specifically, PRAE outperforms all baselines in terms of average AUC. Moreover, both  $\text{PRAE-}\ell_0$  and  $\text{PRAE-}\ell_1$  have a median rank of 2, and each would have a median rank 1 in the absence of the other approach as a competitor.

In the supplemental material, we further demonstrate that our method can accurately identify outliers in the “out-of-sample” setting (Section S2). Box plots indicating the stability of our approach also appear in Section S2. We also provide: reproducibility required technical details (Section S3 and S4), hyperparameters sensitivity evaluation (Section S5), ablation study (Section S6), a deeper analysis of MNIST-S and Fashion MNIST (Section S8), and an analysis of the running time and computational complexity of the method (Section S9).

## 6 Strengths and Limitations

The proposed method provides several advantages compared to a standard autoencoder: (1) It removes outliers along the training process, therefore, can more accurately identify a low-dimensional subspace that represents the normal samples. (2) It provides a reliable metric for curating training data via an “in-sample” fashion anomaly detection. (3) It learns an encoder-decoder mapping that can be used for filtration of anomalies from new unseen samples (“out-of-sample” regime). The success of our method relies on the assumption that normal samples lie near a latent low dimensional manifold, while the outliers are diverse and do not obey such structure. Indeed, there are cases where this assumption does not hold, and our method will not be optimal for identifying the outliers. While robust PCA assumes inliers samples are low rank, our method identifies those as samples that can be “easily” reconstructed by an AE. To prevent the AE from overfitting to outliers, we limit its capacity using a low-dimensional bottleneck. Nonetheless, understanding the properties of an AE from an optimization perspective is an open research direction. A related problem has been studied in supervised learning in recent years [64, 65]. Another limitation of our method is the regularization parameter  $\lambda$ . In our experiments, we kept  $\lambda = 1$  following the intuition provided in section 5.2. Nonetheless, a rigorous scheme for finding the optimal value of  $\lambda$  is an interesting question for future work. Another interesting direction involves incorporating the proposed probabilistic robust mechanism into an attention architecture.

## 7 Conclusion

In this work, we present a novel methodology for anomaly detection. Our method, which we call Probabilistic Robust autoencoder (PRAE), is based on a regularized AE designed to remove samples that do not lie near a low dimensional manifold. Specifically, we multiply each data instance with an approximately binary random variable and add a penalty term to the AE training to encourage sparsity of the number of samples used by the model. We prove an equivalence between the solution of our probabilistic formulation and the solution of an intractable  $\ell_0$  regularized AE. Finally, we demonstrate different properties of the proposed method using extensive simulations. Overall, we obtain several state-of-the-art results on synthetic and real datasets with diverse properties (size, dimensions, and type of features).

## References

- [1] Michael Lenning, Joseph Fortunato, Tai Le, Isaac Clark, Ang Sherpa, Soyeon Yi, Peter Hofsteen, Geethapriya Thamilarasu, Jingchun Yang, Xiaolei Xu, et al. Real-time monitoring and analysis of zebrafish electrocardiogram with anomaly detection. *Sensors*, 18(1):61, 2018.
- [2] Y Bregman, O Lindenbaum, and N Rabin. Array based earthquakes-explosion discrimination using diffusion maps. *Pure and Applied Geophysics*, 178(7):2403–2418, 2021.
- [3] Ashima Chawla, Paul Jacob, Brian Lee, and Sheila Fallon. Bidirectional lstm autoencoder for sequence based anomaly detection in cyber security. *International Journal of Simulation-Systems, Science & Technology*, 2019.
- [4] Christopher M Bishop. Novelty detection and neural network validation. *IEE Proceedings-Vision, Image and Signal processing*, 141(4):217–222, 1994.
- [5] Markus M Breunig, Hans-Peter Kriegel, Raymond T Ng, and Jörg Sander. Lof: identifying density-based local outliers. In *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, pages 93–104, 2000.
- [6] Wen Jin, Anthony KH Tung, and Jiawei Han. Mining top-n local outliers in large databases. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 293–298, 2001.
- [7] Jian Tang, Zhixiang Chen, Ada Wai-Chee Fu, and David W Cheung. Enhancing effectiveness of outlier detections for low density patterns. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 535–548. Springer, 2002.
- [8] Wen Jin, Anthony KH Tung, Jiawei Han, and Wei Wang. Ranking outliers using symmetric neighborhood relationship. In *Pacific-Asia conference on knowledge discovery and data mining*, pages 577–593. Springer, 2006.
- [9] Hans-Peter Kriegel, Peer Kröger, Erich Schubert, and Arthur Zimek. Loop: local outlier probabilities. In *Proceedings of the 18th ACM conference on Information and knowledge management*, pages 1649–1652, 2009.
- [10] Valentino Constantinou. Pynomaly: Anomaly detection using local outlier probabilities (loop). *Journal of Open Source Software*, 3(30):845, 2018.
- [11] Yariv Aizenbud, Amit Bermanis, and Amir Averbuch. Pca-based out-of-sample extension for dimensionality reduction. *arXiv preprint arXiv:1511.00831*, 2015.
- [12] Sridhar Ramaswamy, Rajeev Rastogi, and Kyuseok Shim. Efficient algorithms for mining outliers from large data sets. In *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, pages 427–438, 2000.
- [13] Fabrizio Angiulli and Clara Pizzuti. Fast outlier detection in high dimensional spaces. In *European conference on principles of data mining and knowledge discovery*, pages 15–27. Springer, 2002.

- [14] Amol Ghosh, Srinivasan Parthasarathy, and Matthew Eric Otey. Fast mining of distance-based outliers in high-dimensional datasets. *Data Mining and Knowledge Discovery*, 16(3):349–364, 2008.
- [15] Yunqiang Chen, Xiang Sean Zhou, and Thomas S Huang. One-class svm for learning in image retrieval. In *Proceedings 2001 International Conference on Image Processing (Cat. No. 01CH37205)*, volume 1, pages 34–37. IEEE, 2001.
- [16] Lukas Ruff, Robert Vandermeulen, Nico Goernitz, Lucas Deecke, Shoaib Ahmed Siddiqui, Alexander Binder, Emmanuel Müller, and Marius Kloft. Deep one-class classification. In *International conference on machine learning*, pages 4393–4402. PMLR, 2018.
- [17] Pramuditha Perera, Poojan Oza, and Vishal M Patel. One-class classification: A survey. *arXiv preprint arXiv:2101.03064*, 2021.
- [18] Hanqiu Deng and Xingyu Li. Anomaly detection via reverse distillation from one-class embedding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9737–9746, 2022.
- [19] Yariv Aizenbud and Barak Sober. Non-parametric estimation of manifolds from noisy data. *arXiv preprint arXiv:2105.04754*, 2021.
- [20] Sam T Roweis and Lawrence K Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, 2000.
- [21] Erez Peterfreund, Ofir Lindenbaum, Felix Dietrich, Tom Bertalan, Matan Gavish, Ioannis G Kevrekidis, and Ronald R Coifman. Local conformal autoencoder for standardized data coordinates. *Proceedings of the National Academy of Sciences*, 117(49):30918–30927, 2020.
- [22] Karl Pearson. LIII. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11):559–572, 1901.
- [23] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning internal representations by error propagation. Technical report, California Univ San Diego La Jolla Inst for Cognitive Science, 1985.
- [24] Nathalie Japkowicz, Catherine Myers, Mark Gluck, et al. A novelty detection approach to classification. In *IJCAI*, volume 1, pages 518–523. Citeseer, 1995.
- [25] Yann LeCun et al. Generalization and network design strategies. *Connectionism in perspective*, 19:143–155, 1989.
- [26] Gilad Lerman and Tyler Maunu. An overview of robust subspace recovery. *Proceedings of the IEEE*, 106(8):1380–1410, 2018.
- [27] Jinghui Chen, Saket Sathe, Charu Aggarwal, and Deepak Turaga. Outlier detection with autoencoder ensembles. In *Proceedings of the 2017 SIAM international conference on data mining*, pages 90–98. SIAM, 2017.
- [28] Chong Zhou and Randy C Paffenroth. Anomaly detection with robust deep autoencoders. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 665–674, 2017.

- [29] 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. 2018.
- [30] Yezheng Liu, Zhe Li, Chong Zhou, Yuanchun Jiang, Jianshan Sun, Meng Wang, and Xiangnan He. Generative adversarial active learning for unsupervised outlier detection. *IEEE Transactions on Knowledge and Data Engineering*, 32(8):1517–1528, 2019.
- [31] Yilun Du and Igor Mordatch. Implicit generation and modeling with energy based models. *Advances in Neural Information Processing Systems*, 32, 2019.
- [32] Simao Eduardo, Alfredo Nazábal, Christopher KI Williams, and Charles Sutton. Robust variational autoencoders for outlier detection and repair of mixed-type data. In *International Conference on Artificial Intelligence and Statistics*, pages 4056–4066. PMLR, 2020.
- [33] Dan Hendrycks, Mantas Mazeika, and Thomas Dietterich. Deep anomaly detection with outlier exposure. *arXiv preprint arXiv:1812.04606*, 2018.
- [34] Siqi Wang, Yijie Zeng, Xinwang Liu, En Zhu, Jianping Yin, Chuanfu Xu, and Marius Kloft. Effective end-to-end unsupervised outlier detection via inlier priority of discriminative network. In *NeurIPS*, pages 5960–5973, 2019.
- [35] Sachin Goyal, Aditi Raghunathan, Moksh Jain, Harsha Vardhan Simhadri, and Prateek Jain. Drocc: Deep robust one-class classification. In *International Conference on Machine Learning*, pages 3711–3721. PMLR, 2020.
- [36] Tal Reiss, Niv Cohen, Liron Bergman, and Yedid Hoshen. Panda: Adapting pretrained features for anomaly detection and segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2806–2814, 2021.
- [37] Hadi Hojjati and Narges Armanfard. Dasvdd: Deep autoencoding support vector data descriptor for anomaly detection. *arXiv preprint arXiv:2106.05410*, 2021.
- [38] Dan Hendrycks, Mantas Mazeika, Saurav Kadavath, and Dawn Song. Using self-supervised learning can improve model robustness and uncertainty. *Advances in Neural Information Processing Systems*, 32, 2019.
- [39] Liron Bergman and Yedid Hoshen. Classification-based anomaly detection for general data. *arXiv preprint arXiv:2005.02359*, 2020.
- [40] Lucas Deecke, Lukas Ruff, Robert A Vandermeulen, and Hakan Bilen. Transfer-based semantic anomaly detection. In *International Conference on Machine Learning*, pages 2546–2558. PMLR, 2021.
- [41] Elad Plaut. From principal subspaces to principal components with linear autoencoders. *arXiv preprint arXiv:1804.10253*, 2018.
- [42] Bang Xiang Yong and Alexandra Brintrup. Do autoencoders need a bottleneck for anomaly detection? *arXiv preprint arXiv:2202.12637*, 2022.
- [43] Yutaro Yamada, Ofir Lindenbaum, Sahand Negahban, and Yuval Kluger. Feature selection using stochastic gates. In *International Conference on Machine Learning*, pages 10648–10659. PMLR, 2020.

- [44] Ofir Lindenbaum, Moshe Salhov, Amir Averbuch, and Yuval Kluger. Deep gated canonical correlation analysis. *arXiv preprint arXiv:2010.05620*, 2020.
- [45] Ofir Lindenbaum, Uri Shaham, Jonathan Svirsky, Erez Peterfreund, and Yuval Kluger. Differentiable unsupervised feature selection based on a gated laplacian. *Neural Information Processing Systems (NeurIPS)*, 2021.
- [46] Mayu Sakurada and Takehisa Yairi. Anomaly detection using autoencoders with nonlinear dimensionality reduction. In *Proceedings of the MLSDA 2014 2nd workshop on machine learning for sensory data analysis*, pages 4–11, 2014.
- [47] Stephen Boyd, Stephen P Boyd, and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.
- [48] Gilad Lerman and Tyler Maunu. Fast, robust and non-convex subspace recovery. *Information and Inference: A Journal of the IMA*, 7(2):277–336, 2018.
- [49] Teng Zhang. Robust subspace recovery by tyler’s m-estimator. *Information and Inference: A Journal of the IMA*, 5(1):1–21, 2016.
- [50] 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.
- [51] Zhouchen Lin, Minming Chen, and Yi Ma. The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices. *arXiv preprint arXiv:1009.5055*, 2010.
- [52] Tyler Maunu, Teng Zhang, and Gilad Lerman. A well-tempered landscape for non-convex robust subspace recovery. *Journal of Machine Learning Research*, 20(37), 2019.
- [53] Chieh-Hsin Lai, Dongmian Zou, and Gilad Lerman. Robust subspace recovery layer for unsupervised anomaly detection. *arXiv preprint arXiv:1904.00152*, 2019.
- [54] Yoshinao Ishii and Masaki Takanashi. Low-cost unsupervised outlier detection by autoencoders with robust estimation. *Journal of Information Processing*, 27:335–339, 2019.
- [55] Yue Zhao, Zain Nasrullah, and Zheng Li. Pyod: A python toolbox for scalable outlier detection. *Journal of Machine Learning Research*, 20(96):1–7, 2019.
- [56] Zengyou He, Xiaofei Xu, and Shengchun Deng. Discovering cluster-based local outliers. *Pattern Recognition Letters*, 24(9-10):1641–1650, 2003.
- [57] Jian Tang, Zhixiang Chen, Ada Wai-Chee Fu, and David W Cheung. Enhancing effectiveness of outlier detections for low density patterns. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 535–548. Springer, 2002.
- [58] Hans-Peter Kriegel, Peer Kröger, Erich Schubert, and Arthur Zimek. Outlier detection in axis-parallel subspaces of high dimensional data. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 831–838. Springer, 2009.
- [59] Markus Goldstein and Andreas Dengel. Histogram-based outlier score (hbos): A fast unsupervised anomaly detection algorithm. *KI-2012: Poster and Demo Track*, pages 59–63, 2012.

- [60] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. Isolation forest. In *2008 Eighth IEEE International Conference on Data Mining*, pages 413–422. IEEE, 2008.
- [61] Yue Zhao, Zain Nasrullah, Maciej K Hryniwicki, and Zheng Li. Lscp: Locally selective combination in parallel outlier ensembles. In *Proceedings of the 2019 SIAM International Conference on Data Mining*, pages 585–593. SIAM, 2019.
- [62] Hans-Peter Kriegel, Matthias Schubert, and Arthur Zimek. Angle-based outlier detection in high-dimensional data. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 444–452, 2008.
- [63] Bernhard Schölkopf, John C Platt, John Shawe-Taylor, Alex J Smola, and Robert C Williamson. Estimating the support of a high-dimensional distribution. *Neural computation*, 13(7):1443–1471, 2001.
- [64] Naftali Tishby and Noga Zaslavsky. Deep learning and the information bottleneck principle. In *2015 ieee information theory workshop (itw)*, pages 1–5. IEEE, 2015.
- [65] Basri Ronen, David Jacobs, Yoni Kasten, and Shira Kritchman. The convergence rate of neural networks for learned functions of different frequencies. *Advances in Neural Information Processing Systems*, 32, 2019.
- [66] Grace XY Zheng, Jessica M Terry, Phillip Belgrader, Paul Ryvkin, Zachary W Bent, Ryan Wilson, Solongo B Ziraldo, Tobias D Wheeler, Geoff P McDermott, Junjie Zhu, et al. Massively parallel digital transcriptional profiling of single cells. *Nature communications*, 8(1):1–12, 2017.

## Supplemental Material

Here, we describe the technical details for the experiment performed in Section 6.1. We generate  $N = 10000$  points in  $\mathbb{R}^{200}$  in the following way: first we generate  $N_{in}$  points in  $\mathbb{R}^{200}$  with distribution  $N(0, I_{200})$  where  $I_{200} \in \mathbb{R}^{200 \times 200}$  is the identity matrix. This set of points is orthogonally projected into a random 10 dimensional space and denoted by  $\mathbf{X}_{in}^{high}$ . The orthogonal projection is performed by projecting onto the basis of the column space of a random Gaussian matrix.  $\mathbf{X}_{in}^{high}$  are normalized so that the expectation of the norm is 1. Next, we generate  $\mathbf{X}_{out}$  as random points in  $\mathbb{R}^{200}$  with distribution  $N(0, 1/\sqrt{200}I_{200})$ . This is done so that the expected value of the norm of the outliers will be equal to the expected value of the norm of the inliers. Finally, the dataset is constructed by combining  $\mathbf{X}_{out}, \mathbf{X}_{in}^{high}$  and adding noise  $\sim N(0, \sigma^2)$ . For the results in Table 2 a negligible noise with  $\sigma^2 = 10^{-8}$  was added. In the second experiment a noise of  $\sigma^2 = 10^{-2}$  was added.

## S1 Additional Experimental Details

First, we present the result for the “out-of-sample” setting, in which we split each dataset and use one half for training while the hold-out-set is used to evaluate the performance without additional training. Here, since we don’t have gates from unseen samples, the anomaly score for  $\mathbf{x}_i$  is based on the reconstruction error  $\|\mathbf{x}_i - \hat{\mathbf{x}}_i\|_2^2$ . We compare the performance of each of the algorithms that were compared in the “in-sample” setting. The results are summarized in Table S1. As evident from this experiment, the proposed approach works well in the “out-of-sample” setting. Specifically, the AUC improves on most datasets, and the proposed schemes outperform all baselines in terms of average AUC.

Data set	CBLOF	COF	IForest	SOD	LSCP	HBOS	OC-SVM	DSVDD	RSR-AE	$\ell_{2,1}$ -AE	PRAE- $\ell_0$	PRAE- $\ell_1$
Thyroid	90.92	68.44	<b>97.85</b>	88.41	93.28	<b>95.91</b>	96.13	89.74	75.31	95.50	95.81	<b>95.91</b>
Cardio	84.54	54.52	91.95	67.28	70.2	84.10	58.39	61.89	69.44	<b>94.08</b>	<b>94.58</b>	93.03
Ecoli	81.48	87.77	86.41	92.50	95.60	<b>96.84</b>	<b>99.26</b>	92.07	76.39	49.57	90.82	93.36
Lympho	95.30	98.59	<b>99.06</b>	84.50	98.59	99.98	99.28	80.20	63.90	<b>99.24</b>	93.13	93.84
Pendigits	97.16	52.2	93.35	69.58	50.45	92.69	95.31	78.57	76.02	93.51	<b>98.72</b>	<b>98.93</b>
Yeast	67.39	59.09	76.52	78.89	61.62	82.76	81.70	72.24	49.91	74.37	<b>81.86</b>	<b>83.48</b>
Musk	95.29	54.77	97.48	93.68	92.27	99.28	70.92	80.17	77.76	<b>99.99</b>	<b>98.95</b>	99.74
Mammography	83.94	71.98	86.26	79.67	51.90	83.39	87.48	83.76	82.20	81.59	<b>88.10</b>	<b>88.32</b>
Shuttle	68.75	51.45	<b>99.97</b>	58.28	53.76	98.43	99.14	97.89	71.61	98.81	<b>99.22</b>	99.10
MNIST-S	52.97	52.23	83.37	51.14	51.57	51.36	90.20	80.92	91.29	90.19	<b>93.54</b>	<b>93.31</b>
Fashion-MNIST	70.69	78.11	92.12	88.54	88.12	75.64	88.10	86.66	68.25	89.77	<b>90.32</b>	<b>89.18</b>
PBMC	90.28	92.29	83.37	90.28	91.74	<b>94.35</b>	90.20	80.92	91.29	90.19	<b>93.54</b>	93.31
Credit	NA	NA	95.31	NA	NA	94.83	95.24	63.88	82.69	93.87	<b>95.36</b>	<b>95.65</b>
Average-AUC	81.55	68.45	91.01	78.56	74.92	88.42	88.56	80.68	75.08	88.51	<b>92.76</b>	<b>93.16</b>
Median rank	7.5	11	4.5	9.5	7.5	4.5	4	8	9.5	5	2.5	3

Table S1: Performance comparison with several leading anomaly detection baselines in the “out-of-sample” setting. We present the median AUC over 10 runs.

Next, to evaluate the stability of the proposed approach, we run the method with 10 random initialization on each of the real datasets. In Figure S1 we present box plots indicating the mean, median and 25/75 percentiles of PRAE- $\ell_0$  and PRAE- $\ell_1$  (left and right panels). As evident from this figure, the interquartile range (IQR) of the proposed approach is relatively small for larger datasets (such as Shuttle, Pen, and MNIST-S). Finally, we evaluate the maximal F1 score attained by each method (when varying the anomaly score threshold). The median of this value across ten runs is reported in Table S2.

Data set	CBLOF	COF	IForest	SOD	LSCP	HBOS	OC-SVM	DSVDD	RSR-AE	$\ell_{2,1}$ -AE	PRAE- $\ell_0$	PRAE- $\ell_1$
Thyroid	0.38	0.11	<b>0.61</b>	0.33	0.35	<b>0.53</b>	0.48	0.22	0.15	0.41	0.45	0.43
Cardio	0.50	0.22	0.52	0.33	0.30	0.47	0.59	0.30	0.18	<b>0.64</b>	<b>0.65</b>	0.63
Ecoli	0.50	0.67	0.74	0.50	<b>0.80</b>	0.42	<b>0.81</b>	0.70	0.17	0.09	0.76	0.75
Lympho	0.50	0.63	<b>0.82</b>	0.62	0.73	<b>0.92</b>	0.87	0.28	0.57	0.82	0.62	0.71
Pendigits	0.45	0.08	0.37	0.10	0.06	0.35	0.41	0.18	0.12	0.32	<b>0.63</b>	<b>0.66</b>
Yeast	0.20	0.19	0.34	0.18	0.23	0.29	0.34	0.15	0.33	0.21	<b>0.39</b>	<b>0.40</b>
Musk	0.55	0.11	0.71	0.35	0.55	0.84	0.47	0.70	0.80	<b>0.98</b>	0.80	<b>0.87</b>
Mammography	0.28	0.18	0.25	0.17	0.05	0.18	0.29	0.29	0.04	0.28	<b>0.30</b>	<b>0.30</b>
Shuttle	0.30	0.14	0.95	0.20	0.14	0.96	<b>0.96</b>	0.51	0.14	<b>0.94</b>	0.93	0.92
MNIST-S	0.43	0.47	0.38	0.31	0.61	0.36	0.43	0.28	0.42	0.44	<b>0.53</b>	<b>0.65</b>
Fashion-MNIST	0.52	0.27	0.38	0.24	0.34	0.22	0.51	0.29	0.34	0.55	<b>0.62</b>	<b>0.59</b>
Average-F1	0.42	0.28	0.55	0.31	0.38	0.50	0.56	0.36	0.29	0.52	<b>0.61</b>	<b>0.63</b>
Median rank	7	10	5	10	8	6	4	9	10	5	3	2

Table S2: Performance comparison with several leading anomaly detection baselines. We present the median maximal F1 score (where the maximum is computed for each run).

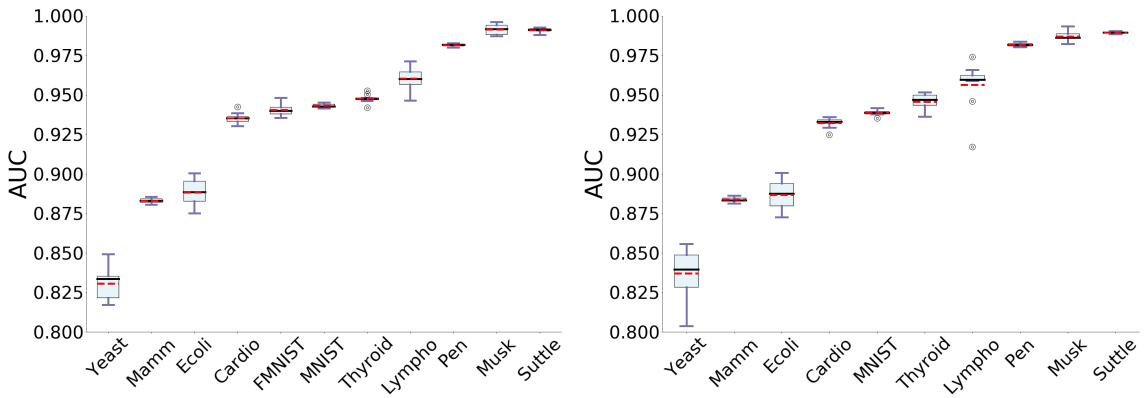


Figure S1: Box plots presenting the AUC of the proposed approach on several real datasets. The top and bottom plots represent PRAE- $\ell_0$  and PRAE- $\ell_1$  respectively. We use 10 different random initializations on each dataset and compute the AUC after convergence.

## S2 Datasets

All real world datasets analyzed in this study are publicly available. Most datasets can be directly downloaded from <sup>1</sup>. MNIST and Fashion-MNIST can be easily obtained using python from “keras” package. The Purified populations of peripheral blood monocytes (PBMCs) is a Single-cell RNA sequencing (scRNA-seq) data. It was collected by [66] and contains more than 90,000 cells with 32,738 genes. We randomly sample 6,000 CD34 cells as our normal samples, and add 300 cells from the remaining 8 cell types. Since typically in scRNA-seq there is some level of false annotations, we design this example to evaluate the capabilities of PRAE in curating a set of CD34 from contamination by other cell types.

## S3 Baselines and Hyperparameters

In the following section, we describe all baselines and hyperparameters used for the evaluation of anomaly detection using real and synthetic data. We train our proposed AE with an encoder-decoder pair with five hidden layers each of size 10; the hidden dimension is 1 (this might not be optimal but worked well across most datasets). For the high dimensional datasets (PBMC, Fashion-MNIST) the hidden layers are of size 100. We use the heuristic proposed in Section 6.2 to tune the regularization parameter to  $\lambda = 1 < ME$ . We use Adam optimizer with a learning

<sup>1</sup><http://odds.cs.stonybrook.edu/>

rate of  $N \cdot 10^{-6}$  where  $N$  is the number of samples in the dataset.

**Clustering-Based Local Outlier Factor (CBLOF)** [56] is a proximity-based method that relies on clustering to define an anomaly score for each sample.

**Angle-Based Outlier Detection (ABOD)** [62] uses vectors defined between pairs of points to identify outliers by comparing the angles between the different vectors.

**Connectivity-Based Outlier Factor(COF)** [57] relies on proximity between samples to identify outliers.

**Isolation Forest (IForest)** [60] seeks the minimum number of splits required to isolate each sample. Then, an ensemble of trees are used to define an anomaly score.

**Subspace Outlier Detection (SOD)** [58] identifies anomalies as samples that deviate significantly from the subspace spanned by a subspace defined based on neighbors of point.

**Locally Selective Combination of Parallel Outlier Ensembles (LSCP)** [61] is an ensemble method that uses multiple local subspaces to identify outliers.

**Histogram-based Outlier Score (HBOS)** [59] is a probabilistic approach that creates histograms to identify anomalies as samples with low density.

**One-Class Support Vector Machines (OC-SVM)** [63] uses support vectors to identify the margins of the normal part of the data. Here, we use a Gaussian kernel to capture the non-linearity of this method.

**Deep One-Class Classification (Deep-SVDD)** [16] extends OC-SVM by introducing a NN to model the decision boundary of the normal data. Here, we evaluated several architectures for DSVDD and selected the one that leads to the smallest validation error. The method is optimized with Adam and the number of epochs in 100.

**Robust deep autoencoders ( $\ell_{2,1} - AE$ )** [28] uses an  $\ell_{2,1}$  to regularize an AE that attempts to reconstruct the data while removing outliers. The approach is explained in the main text. We use an AE with the same architecture as the proposed approach in all examples.

**Ensemble of Autoencoders (RandNet)** [27] In the absence of an implementation of the method, we report the results presented by the authors. The method relies on an aggregated of an ensemble of AEs. The AUC of the method for 6 out of the to 13 analyzed datasets are reported in Table S3. Our method outperforms Randnet and the majority of these datasets.

dataset	RandNet	Rank
Thyroid	90.42	6
Cardio	92.87	3
Ecoli	85.42	9
Lympho	99.06	2
Pendigits	93.44	5
Yeast	82.95	3

Table S3: AUC results of RandNet as reported by the authors along with the rank compared to all existing baselines. These were omitted from the table in the main text since the datasets only partially overlap.

## S4 Sensitivity to Hyperparameters

In the following experiment, we evaluate the sensitivity of the method to hyperparameters. Towards this goal, we run PRAE on the ‘Yeast’ and ‘Musk’ datasets for various values of  $\lambda$  and different learning rates. These datasets where arbitrarily chosen from the datasets in Table

2. For each value of the parameters we repeat the experiment ten times and report the median AUC. In Figure S2 we present heatmaps with the median AUC values for each set of parameters used in the evaluation. As demonstrated by this figure, PRAE is relatively stable to both of these hyperparameters, and the overall AUC varies by less than 10% across all evaluated values of  $\lambda$  and the learning rate. As expected, if the value of  $\lambda$  is small, the performance of the model deteriorates. This could happen if the model removes too many inliers from the objective at an early stage of training, thus "hurting" the ability to distinguish between inliers and outliers.

Another hint for the stability of the algorithm to the hyperparameter  $\lambda$  is the fact that we used the same  $\lambda$  for all experiments in Table 2 and did not choose a different value for each experiment.

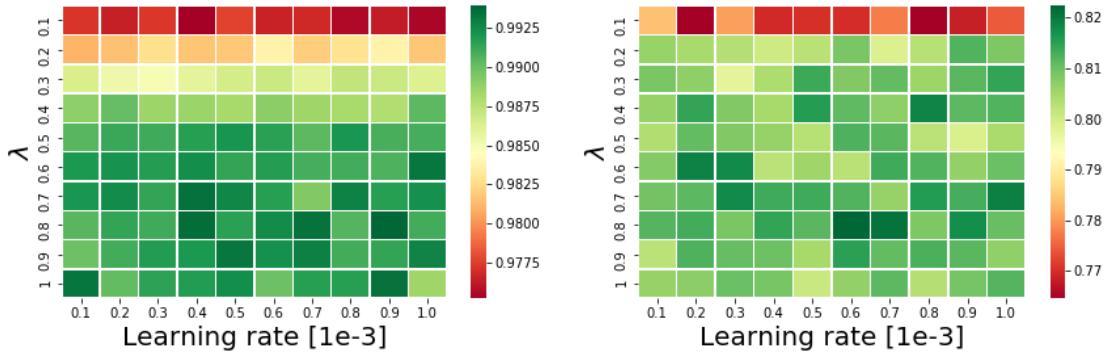


Figure S2: Heatmap presenting the AUC of the proposed approach on for several values of  $\lambda$  and the learning rate. The left plot indicates performance on ‘Musk’ data, while the right plot is based on the ‘Yeast’ data.

## S5 Ablation Study

In this section, we perform an ablation study to evaluate the influence of each element of our algorithm. We focus on the ‘Yeast’ and ‘Musk’ datasets, and compare PRAE- $\ell_0$ , and PRAE- $\ell_1$ , to the following variants:

- **AE**- a standard AE with no regularization, anomalies are identified using the reconstruction error.
- **DRAE- $\ell_1$** - a variant of PRAE, but with deterministic gate values, and a standard  $\ell_1$  regularizer.
- **PRAE ( $\lambda = 0$ )**- a variant of PRAE, but with a regularization term= 0.

As demonstration in Table S5 the proposed probabilistic regularization leads to improved identification of outliers compared with all variants of the method. This empirical results, suggests that removing outliers throughout training with the stochastic gates can lead to more reliable identification of outliers using AEs.

## S6 Synthetic Swiss Roll

In the synthetic example described in Section 6.3, we consider a “narrow swiss-roll”, with 1000 points uniformly sampled from  $[3\pi/2, 9\pi/2] \times [0, 0.1]$ , and embedded into  $\mathbb{R}^3$  using  $(t, h) \rightarrow$

dataset	AE	DRAE- $\ell_1$	PRAE ( $\lambda = 0$ )	PRAE- $\ell_0$	PRAE- $\ell_1$
Yeast	74.05	77.26	77.18	83.37	83.95
Musk	95.61	96.81	88.63	99.17	98.61

Table S4: Ablation study. Comparing the proposed schemes PRAE- $\ell_0$ , and PRAE- $\ell_1$ , to three other variants. We use the same architecture, optimizer and compare to AE, DRAE- $\ell_1$  and PRAE ( $\lambda = 0$ ) all explained above.

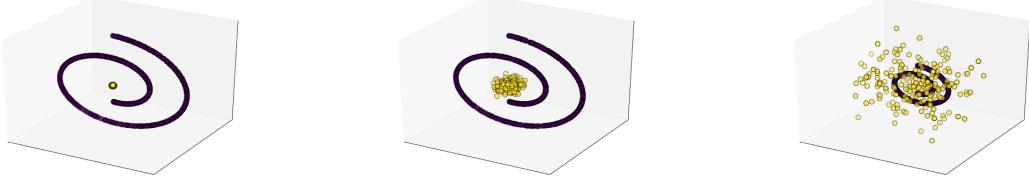


Figure S3: Examples of the synthetic narrow Swiss roll (blue) with Gaussian outliers (yellow), generated using different values of  $\sigma^2 \in 0.1, 1, 10$ .

$(t \cos(t), h, t \sin(t))$ . Then, we generate additional 200 “outliers” sampled from  $N(0, \sigma^2 I_3)$ , where  $I_3 \in \mathbb{R}^{3 \times 3}$  is the identity matrix. In Figure S3 we present examples of Swiss rolls with anomalies generated using different values of  $\sigma$ . In this example, we use a NN with hidden layers of size 512, 256, 128, 64, 32, and the latent space has two neurons. Since the energy of the data varies substantially across samples, we used a normalized reconstruction loss for training all AE-based methods. Specifically, we normalize the reconstruction error of each sample by the  $\ell_2$  norm of the sample.

## S7 MNIST-S and Fashion MNIST

In this section, we describe the more experiments performed using MNIST-S and Fashion MNIST.

### S7.1 Small MNIST Dataset (MNIST-S)

MNIST-S was proposed in [28] for anomaly detection. To construct MNIST-S, we mix 4859 nominal instances of the digit ‘4’ with 265 anomalies randomly sampled from all other digits. Following [28], we use a linear AE with one hidden layer of size 24. Evaluation of the AUC of our method compared to all baselines appears in Table 3. This example appears to be especially challenging for density/distance-based baselines; we believe that this is due to the relatively high dimensionality of this data. In Figure S4, we present the 25 most *inlaying* images (left panel), and the 25 most *outlying* images (center panel) as identified PRAE- $\ell_0$ . The identified inliers share a standard “simple” structure of the digit ‘4’. On the other hand, most identified outliers are indeed of different digits, except for one example, which is somewhat of an unusual instance of the digit ‘4’.

To further assess the performance of the different baselines on MNIST-S we present the 25 most *inlaying* images and the 25 most *outlying* images as identified by the different baselines. These results are presented in Figures S5 and S6.

Next, we repeat the MNIST-S experiment using different classes as the representatives for the normal samples. Specifically, we mix 4859 nominal instances of the digit  $C \in \{0, 1, \dots, 9\}$

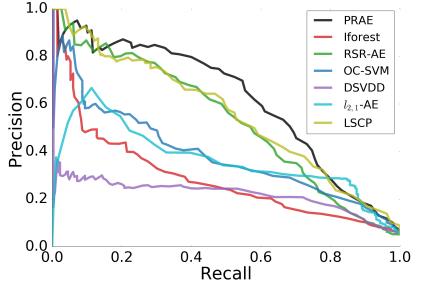
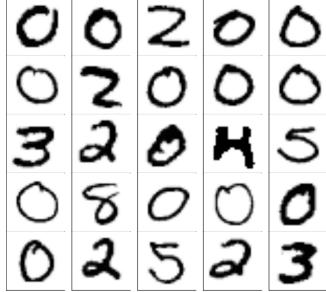
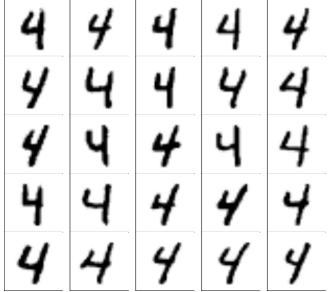


Figure S4: 25 most inlaying/outlying (left/center) MNIST-S images as identified by PRAE. Right- precision vs. recall curves for leading baselines on the MNIST-S data.

with 265 anomalies randomly sampled from all other digits. We compare the AUC of PRAE to two leading baselines across all classes. As indicated in Table S5, the proposed approach leads to more accurate outlier identification compared to the leading baselines across most of the classes.

Class	Iforest	OC-SVM	PRAE- $\ell_0$	PRAE- $\ell_1$
0	92.34	93.34	93.38	93.84
1	99.16	98.2	98.21	98.24
2	69.73	75.72	81.12	81.18
3	78.02	81.27	83.42	83.51
4	86.66	88.21	88.43	88.39
5	73.49	74.79	83.94	83.96
6	87.75	92.17	93.62	93.71
7	90.49	90.79	91.77	91.81
8	82.87	82.47	83.31	83.34
9	86.99	91.46	93.24	93.33
T-shirt/top	90.45	90.62	90.65	90.84
Trouser	97.77	97.83	97.96	97.90
Pullover	86.26	85.27	87.24	86.66
Dress	93.7	94.56	95.28	95.14
Coat	91.95	91.17	92.47	92.15
Sandal	92.12	91.58	91.14	91.15
Shirt	80.48	80.25	80.86	80.88
Sneaker	98.14	98.24	98.23	98.26
Bag	87.14	84.42	84.49	85.58
Ankle boot	97.65	98.16	97.95	97.91
<b>Average</b>	88.15	89.03	<b>90.34</b>	<b>90.39</b>

Table S5: Performance comparison with two leading baselines (Iforest and OC-SVM) on the MNIST-S and Fashion MNIST datasets. The top ten rows correspond to different classes in MNIST; each row indicates the class label used to define the normal samples. The bottom ten rows correspond to different classes in Fashion MNIST.

## S7.2 Fashion MNIST

To evaluate the ability to identify outliers in Fashion MNIST, we mix 5000 nominal instances from the randomly selected 'Coat' class with 300 anomalies sampled from all other fashion items. Evaluation of the AUC of our method compared to all baselines appears in Table 3. Next, we repeat this experiment using other classes as the majority/normal samples. In Table S5 we present the AUC of PRAE and two leading baselines across all classes. As evident from our result, the proposed approach outperforms leading baselines across most classes of Fashion MNIST.

## S8 Running Time and Platform Details

All the experiments were run on a server with an Intel(R) Xeon(R) CPU E5-2620 v3 @ 2.40GHz CPU and one GeForce GTX 1080 GPU with 8GB of memory. Our method scales like a standard autoencoder, and every iteration requires  $\mathcal{O}(M + N)$  updates, where  $M$  is the number of parameters in the network, and  $N$  are the additional per sample parameters that serve as the anomaly score. Since modern NN are typically overparametrized, we argue that the additional  $N$  parameters do not limit the use of our method since typically  $M > N$ . Across all examples used in this paper, a single run of our method does not take more than several minutes for the low-dimensional datasets and hours for the high-dimensional ones.

<u>SOD-Normal</u>	<u>SOD-Anomaly</u>																																																		
<table border="1"> <tbody> <tr><td>1</td><td>4</td><td>1</td><td>4</td><td>1</td></tr> <tr><td>4</td><td>1</td><td>4</td><td>4</td><td>4</td></tr> <tr><td>4</td><td>4</td><td>4</td><td>4</td><td>4</td></tr> <tr><td>4</td><td>1</td><td>4</td><td>4</td><td>4</td></tr> <tr><td>4</td><td>4</td><td>4</td><td>4</td><td>4</td></tr> </tbody> </table>	1	4	1	4	1	4	1	4	4	4	4	4	4	4	4	4	1	4	4	4	4	4	4	4	4	<table border="1"> <tbody> <tr><td>2</td><td>2</td><td>8</td><td>5</td><td>4</td></tr> <tr><td>3</td><td>9</td><td>5</td><td>2</td><td>2</td></tr> <tr><td>8</td><td>1</td><td>3</td><td>3</td><td>4</td></tr> <tr><td>5</td><td>3</td><td>8</td><td>4</td><td>4</td></tr> <tr><td>2</td><td>7</td><td>2</td><td>2</td><td>8</td></tr> </tbody> </table>	2	2	8	5	4	3	9	5	2	2	8	1	3	3	4	5	3	8	4	4	2	7	2	2	8
1	4	1	4	1																																															
4	1	4	4	4																																															
4	4	4	4	4																																															
4	1	4	4	4																																															
4	4	4	4	4																																															
2	2	8	5	4																																															
3	9	5	2	2																																															
8	1	3	3	4																																															
5	3	8	4	4																																															
2	7	2	2	8																																															
<u>LSCP-Normal</u>	<u>LSCP-Anomaly</u>																																																		
<table border="1"> <tbody> <tr><td>4</td><td>4</td><td>4</td><td>4</td><td>4</td></tr> <tr><td>4</td><td>4</td><td>4</td><td>4</td><td>4</td></tr> <tr><td>4</td><td>4</td><td>4</td><td>4</td><td>4</td></tr> <tr><td>4</td><td>4</td><td>4</td><td>4</td><td>4</td></tr> <tr><td>4</td><td>4</td><td>4</td><td>4</td><td>4</td></tr> </tbody> </table>	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	<table border="1"> <tbody> <tr><td>3</td><td>3</td><td>5</td><td>2</td><td>0</td></tr> <tr><td>2</td><td>0</td><td>3</td><td>2</td><td>8</td></tr> <tr><td>4</td><td>4</td><td>2</td><td>3</td><td>3</td></tr> <tr><td>7</td><td>2</td><td>2</td><td>0</td><td>7</td></tr> <tr><td>3</td><td>5</td><td>6</td><td>2</td><td>8</td></tr> </tbody> </table>	3	3	5	2	0	2	0	3	2	8	4	4	2	3	3	7	2	2	0	7	3	5	6	2	8
4	4	4	4	4																																															
4	4	4	4	4																																															
4	4	4	4	4																																															
4	4	4	4	4																																															
4	4	4	4	4																																															
3	3	5	2	0																																															
2	0	3	2	8																																															
4	4	2	3	3																																															
7	2	2	0	7																																															
3	5	6	2	8																																															
<u>HBOS-Normal</u>	<u>HBOS-Anomaly</u>																																																		
<table border="1"> <tbody> <tr><td>4</td><td>4</td><td>4</td><td>4</td><td>4</td></tr> <tr><td>4</td><td>4</td><td>4</td><td>4</td><td>4</td></tr> <tr><td>4</td><td>4</td><td>4</td><td>4</td><td>4</td></tr> <tr><td>4</td><td>4</td><td>4</td><td>4</td><td>4</td></tr> <tr><td>4</td><td>4</td><td>4</td><td>4</td><td>4</td></tr> </tbody> </table>	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	<table border="1"> <tbody> <tr><td>0</td><td>0</td><td>2</td><td>0</td><td>9</td></tr> <tr><td>0</td><td>3</td><td>2</td><td>4</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>0</td><td>4</td><td>0</td></tr> <tr><td>3</td><td>2</td><td>2</td><td>0</td><td>4</td></tr> <tr><td>0</td><td>3</td><td>2</td><td>0</td><td>0</td></tr> </tbody> </table>	0	0	2	0	9	0	3	2	4	0	0	0	0	4	0	3	2	2	0	4	0	3	2	0	0
4	4	4	4	4																																															
4	4	4	4	4																																															
4	4	4	4	4																																															
4	4	4	4	4																																															
4	4	4	4	4																																															
0	0	2	0	9																																															
0	3	2	4	0																																															
0	0	0	4	0																																															
3	2	2	0	4																																															
0	3	2	0	0																																															
<u>ABOD-Normal</u>	<u>ABOD-Anomaly</u>																																																		
<table border="1"> <tbody> <tr><td>1</td><td>1</td><td>4</td><td>4</td><td>4</td></tr> <tr><td>4</td><td>4</td><td>4</td><td>1</td><td>4</td></tr> <tr><td>4</td><td>4</td><td>4</td><td>4</td><td>1</td></tr> <tr><td>1</td><td>4</td><td>4</td><td>4</td><td>4</td></tr> <tr><td>4</td><td>1</td><td>4</td><td>4</td><td>4</td></tr> </tbody> </table>	1	1	4	4	4	4	4	4	1	4	4	4	4	4	1	1	4	4	4	4	4	1	4	4	4	<table border="1"> <tbody> <tr><td>4</td><td>4</td><td>2</td><td>5</td><td>4</td></tr> <tr><td>5</td><td>4</td><td>4</td><td>2</td><td>0</td></tr> <tr><td>3</td><td>4</td><td>2</td><td>4</td><td>4</td></tr> <tr><td>4</td><td>4</td><td>4</td><td>4</td><td>3</td></tr> <tr><td>4</td><td>4</td><td>4</td><td>2</td><td>4</td></tr> </tbody> </table>	4	4	2	5	4	5	4	4	2	0	3	4	2	4	4	4	4	4	4	3	4	4	4	2	4
1	1	4	4	4																																															
4	4	4	1	4																																															
4	4	4	4	1																																															
1	4	4	4	4																																															
4	1	4	4	4																																															
4	4	2	5	4																																															
5	4	4	2	0																																															
3	4	2	4	4																																															
4	4	4	4	3																																															
4	4	4	2	4																																															

Figure S5: 25 most inlaying/outlying (left/right) MNIST-S images as identified by different baseline algorithms.

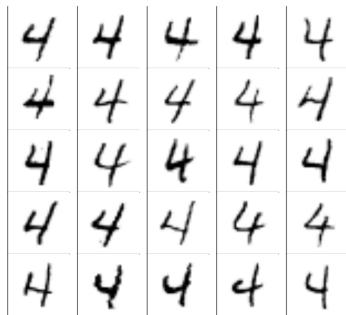
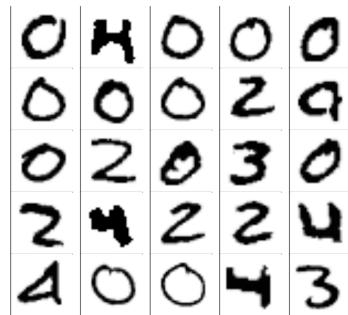
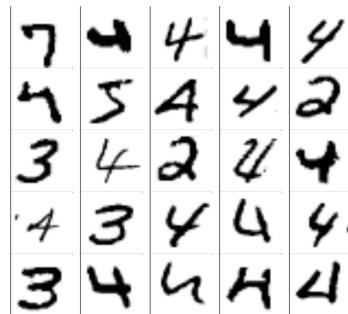
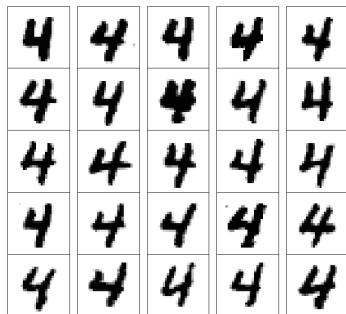
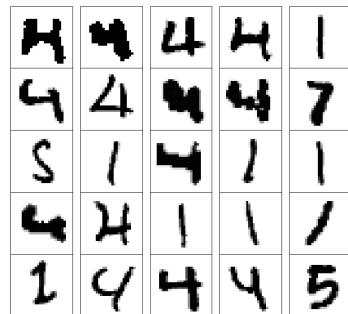
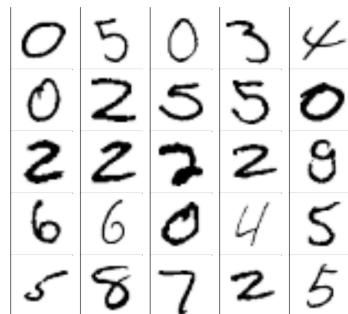
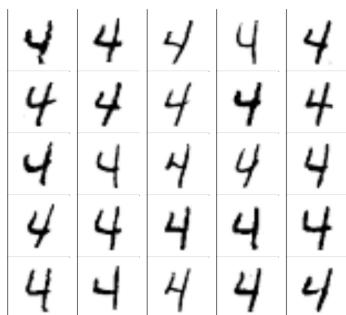
OC-SVM-NormalOC-SVM-AnomalyDSVDD-NormalDSVDD-Anomaly $\ell_{2,1}$ -AE-Normal $\ell_{2,1}$ -AE-AnomalyRSR-AE-NormalRSR-AE-AnomalyIForest-NormalIForest-Anomaly

Figure S6: 25 most inlaying/outlying (left/right) MNIST-S images as identified by different baselines