Deep Semi-Supervised Anomaly Detection

Lukas Ruff[1] **Robert A. Vandermeulen**[2]
Nico Görnitz[1] **Alexander Binder**[3] **Emmanuel Müller**[4]
Klaus-Robert Müller[1 5 6] **Marius Kloft**[2 7]
[1]Technical University of Berlin, Berlin, Germany
[2]Technical University of Kaiserslautern, Kaiserslautern, Germany
[3]Singapore University of Technology & Design, Singapore
[4]b-it Bonn-Aachen, Bonn, Germany
[5]Korea University, Seoul, Republic of Korea
[6]Max Planck Institute for Informatics, Saarbrücken, Germany
[7]University of Southern California, Los Angeles, United States
{lukas.ruff, nico.goernitz, klaus-robert.mueller}@tu-berlin.de
alexander_binder@sutd.edu.sg mueller@bit.uni-bonn.de
{vandermeulen, kloft}@cs.uni-kl.de

Abstract

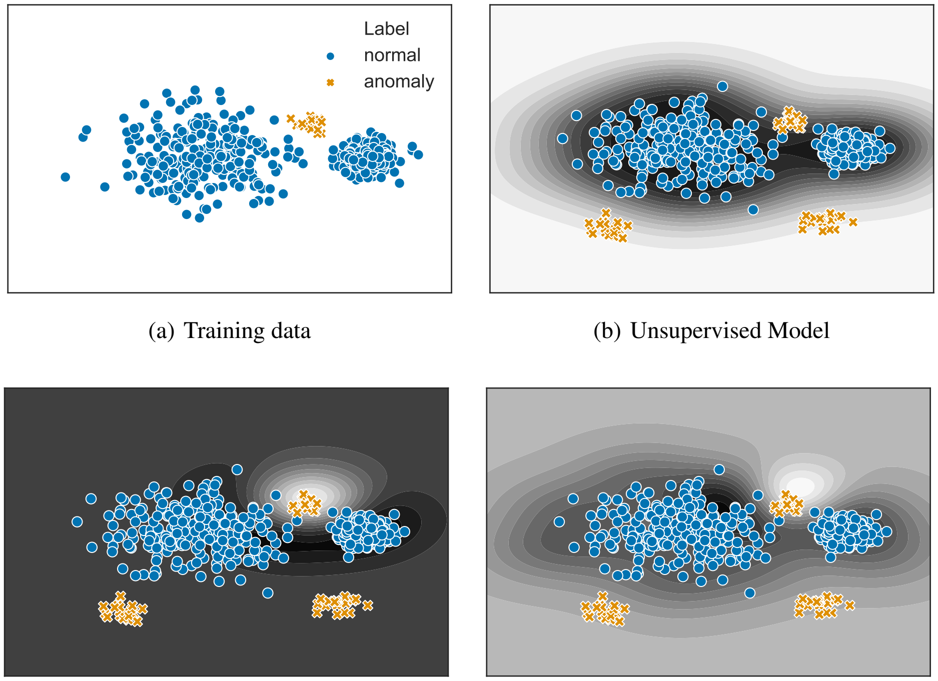
Deep approaches to anomaly detection have recently shown promising results over shallow approaches on high-dimensional data. Typically anomaly detection is treated as an unsupervised learning problem. In practice however, one may have—in addition to a large set of unlabeled samples—access to a small pool of labeled samples, e.g. a subset verified by some domain expert as being normal or anomalous. Semi-supervised approaches to anomaly detection make use of such labeled data to improve detection performance. Few deep semi-supervised approaches to anomaly detection have been proposed so far and those that exist are domain-specific. In this work, we present *Deep SAD*, an end-to-end methodology for deep semi-supervised anomaly detection. Using an information-theoretic perspective on anomaly detection, we derive a loss motivated by the idea that the entropy for the latent distribution of normal data should be lower than the entropy of the anomalous distribution. We demonstrate in extensive experiments on MNIST, Fashion-MNIST, and CIFAR-10 along with other anomaly detection benchmark datasets that our approach is on par or outperforms shallow, hybrid, and deep competitors, even when provided with only few labeled training data.

1 Introduction

Anomaly detection (AD) [10, 51] is the task of identifying unusual samples in data. Because this task lacks a supervised learning objective AD methods typically formulate an unsupervised problem to find a "compact" description of the "normal" class. In one-class classification [44, 60, 65, 56] for example the aim is to find a set of small measure which contains most of the data, and samples that deviate from this description are deemed anomalous. Shallow anomaly detectors such as the One-Class SVM (OC-SVM) [60], Support Vector Data Description (SVDD) [65], Isolation Forest (IF) [38], or Kernel Density Estimator (KDE) [50, 31, 68] often require manual feature engineering to be effective on high-dimensional data and are limited in their scalability to large datasets. These limitations have sparked great interest in developing novel unsupervised *deep learning* approaches to AD, a line of research which has already shown promising results [58, 19, 72, 12, 56, 17, 22, 25].

Unlike the standard AD setting, in many real-world applications one may also have access to some verified (i.e., labeled) normal or anomalous examples in addition to the unlabeled data. Such samples

Preprint. Under review.



(a) Training data

(b) Unsupervised Model

(c) Supervised Model

(d) Semi-Supervised Model (Ours)

Figure 1: The need for semi-supervised AD methods: We consider a setting with only one known anomaly class (orange) at training time (illustrated in (a)) and two new unknown anomaly classes appearing at testing time (bottom left and bottom right in (b), (c), and (d)). The purely unsupervised method (shown in (b)) ignores the known anomalies, which are deemed normal. The purely supervised approach (shown in (c)) overfits to only the previously seen anomalies but fails to generalize to the novel anomalies. Our semi-supervised approach (shown in (d)) strikes a balance.

could be hand labeled by a domain expert, for instance. Unsupervised approaches to AD ignore this valuable information whereas supervised approaches can overfit the training data and fail to generalize to out-of-distribution anomalies. Figure 1 illustrates this situation with a toy example.

Semi-supervised AD [69, 39, 8, 45, 23] aims to bridge the gap between unsupervised AD and supervised learning. These approaches do not assume a common pattern among the “anomaly class” and thus do not impose the typical cluster assumption semi-supervised classifiers build upon [75, 11]. Instead, semi-supervised approaches to AD aim to find a “compact description” of the data while also correctly classifying the labeled instances [8, 23]. Because of this, semi-supervised AD methods do not overfit to the labeled anomalies and generalize well to novel anomalies [23]. Existing work on *deep* semi-supervised learning has almost exclusively focused on classification [34, 53, 48, 16, 49]; only a few deep semi-supervised approaches have been proposed for AD and those tend to be domain or data-type specific [20, 35, 42].

In this work, we present *Deep SAD* (Deep Semi-Supervised Anomaly Detection), an end-to-end deep method for semi-supervised AD. Deep SAD is a generalization of our recently introduced Deep SVDD [56] to include labeled data. We show that our approach can be understood in information-theoretic terms as learning a latent distribution of low entropy for the normal data, with the anomalous distribution having a heavier tailed, higher entropy distribution. To do this we formulate an information-theoretic perspective on deep learning for AD.

2 An Information-theoretic Perspective on Deep Anomaly Detection

The study of the theoretical foundations of deep learning is an active and ongoing research effort [43, 66, 14, 18, 46, 52, 73, 1, 5, 6, 70, 36]. One strong line of research that has emerged is rooted in information theory [62].

In the supervised setting where one has input variable X, latent variable Z (e.g., the final layer of a deep network), and output variable Y (i.e., the label), the well-known *Information Bottleneck* principle [67, 66, 63, 2, 59] is an explanation of representation learning as a trade-off between finding a minimal compression Z of the input X while retaining the informativeness of Z for predicting the label Y. Put formally: supervised deep learning seeks to minimize the mutual information $\mathcal{I}(X; Z)$ between the input X and the latent representation Z while maximizing the mutual information $\mathcal{I}(Z; Y)$ between Z and the task Y, i.e.

$$\min_{p(z|x)} \quad \mathcal{I}(X; Z) - \alpha\, \mathcal{I}(Z; Y), \tag{1}$$

where $p(z|x)$ is modeled by a deep network and the hyperparameter $\alpha > 0$ controls the trade-off between compression (i.e., complexity) and classification accuracy.

For unsupervised deep learning, due to the absence of labels Y and thus the lack of an obvious task, other information-theoretic learning principles have been formulated. Of these, the *Infomax* principle [37, 7, 27] is one of the most prevalent and widely used principles. In contrast to (1), the objective of Infomax is to *maximize* the mutual information $\mathcal{I}(X; Z)$ between the data X and its representation Z

$$\max_{p(z|x)} \quad \mathcal{I}(X; Z) + \beta\, \mathcal{R}(Z). \tag{2}$$

This is typically done using some additional constraint or regularization $\mathcal{R}(Z)$ on the representation Z with hyperparameter $\beta > 0$ to obtain statistical properties desired for some specific downstream task. Examples in which the Infomax principle has been applied have a long history and include unsupervised tasks such as independent component analysis [7], clustering [64, 30], generative modeling [13, 28, 74, 3], and unsupervised representation learning in general [27].

We observe that the Infomax principle has also been implicitly applied in previous deep representations for AD. For example autoencoding models [57, 26], which make up the predominant class of approaches to deep AD [24, 58, 4, 19, 72, 12, 9], can be understood as implicitly maximizing the mutual information $\mathcal{I}(X; Z)$ via the reconstruction objective under some regularization of the latent code Z. Choices for regularization include sparsity [40], the distance to some prior latent distribution, e.g. measured via the KL divergence [33, 55], an adversarial loss [41], or simply a bottleneck in dimensionality. Such restrictions for AD share the idea that the latent representation of the normal data should be in some sense “compact”.

As illustrated in Figure 1, a supervised approach to AD only learns to recognize anomalies similar to those seen in training. However, anything not normal is by definition an anomaly and there is no explicit distribution of the “anomaly class”. This makes supervised learning principles such as (1) ill-defined for AD. We instead build upon principle (2) to derive a deep method for semi-supervised AD, where we include the label information Y through a novel representation learning regularizer $\mathcal{R}(Z) = \mathcal{R}(Z; Y)$ that is based on entropy.

3 Deep Semi-supervised Anomaly Detection

In the following, we introduce *Deep SAD*, a deep method for semi-supervised AD. To formulate our objective, we first briefly review the unsupervised Deep SVDD method [56] and show its connection to entropy minimization. We then generalize the method to the semi-supervised AD setting.

3.1 Unsupervised Deep SVDD

For input space $\mathcal{X} \subseteq \mathbb{R}^D$ and output space $\mathcal{Z} \subseteq \mathbb{R}^d$, let $\phi(\cdot\,; \mathcal{W}) : \mathcal{X} \to \mathcal{Z}$ be a neural network with L hidden layers and corresponding set of weights $\mathcal{W} = \{\boldsymbol{W}^1, \dots, \boldsymbol{W}^L\}$. The objective of Deep SVDD is to train a neural network ϕ to learn a transformation that minimizes the volume of a data-enclosing hypersphere in output space $\mathcal{Z}$ centered on a predetermined point $\boldsymbol{c}$. Given n (unlabeled) training samples $\boldsymbol{x}_1, \dots, \boldsymbol{x}_n \in \mathcal{X}$, the *One-Class Deep SVDD* objective is defined as:

$$\min_{\mathcal{W}} \quad \frac{1}{n} \sum_{i=1}^{n} \|\phi(\boldsymbol{x}_i; \mathcal{W}) - \boldsymbol{c}\|^2 + \frac{\lambda}{2} \sum_{\ell=1}^{L} \|\boldsymbol{W}^\ell\|_F^2. \tag{3}$$

The Deep SVDD penalizes the mean squared distance of the mapped data points to the center $\boldsymbol{c}$ of the sphere. This forces the network to extract those common factors of variation which are most stable within a dataset. As a consequence, normal data points tend to get mapped near the hypersphere center, whereas anomalies are mapped further away [56]. The second term is a weight decay regularizer on the network weights $\mathcal{W}$ with $\lambda > 0$, where $\|\cdot\|_F$ denotes the Frobenius norm.

The unsupervised Deep SVDD can be optimized via SGD using backpropagation. For initialization, the authors first pre-train an autoencoder and then initialize the network ϕ with the converged weights of the encoder. After initializing the network weights $\mathcal{W}$, the hypersphere center $\boldsymbol{c}$ is fixed as the mean of the network representations obtained from an initial forward pass on the training data [56].

The anomaly score of a test point $\boldsymbol{x}$ finally is given by its distance to the center of the hypersphere:

$$s(\boldsymbol{x}) = \|\phi(\boldsymbol{x}; \mathcal{W}^*) - \boldsymbol{c}\|, \tag{4}$$

where $\mathcal{W}^*$ are the network weights of a trained model.

3.2 Deep SVDD and Entropy Minimization

We now show that Deep SVDD may not only be understood in terms of minimum volume estimation [61], but also in terms of entropy minimization over the latent distribution. For a (continuous) latent random variable Z with pdf $p(\boldsymbol{z})$ and support $\mathcal{Z} \subseteq \mathbb{R}^d$, its (differential) entropy is given by

$$\mathcal{H}(Z) = \mathbb{E}[-\log p(Z)] = -\int_{\mathcal{Z}} p(\boldsymbol{z}) \log p(\boldsymbol{z}) \, \mathrm{d}\boldsymbol{z}. \tag{5}$$

Assuming Z has finite covariance Σ, it follows that

$$\mathcal{H}(Z) \leq \frac{1}{2} \log(\det 2\pi e \Sigma) = \frac{1}{2} \log((2\pi e)^d \det \Sigma) \tag{6}$$

with equality if and only if Z is jointly Gaussian [15]. Thus, if Z follows an isotropic Gaussian, $Z \sim N(\boldsymbol{\mu}, \sigma^2 I)$, with $\sigma > 0$, then

$$\mathcal{H}(Z) = \frac{1}{2} \log((2\pi e)^d \det \sigma^2 I) = \frac{1}{2} \log((2\pi e \sigma^2)^d \cdot 1) = \frac{d}{2}(1 + \log(2\pi\sigma^2)) \propto \log \sigma^2, \tag{7}$$

i.e. for a fixed dimensionality d, the entropy of Z is proportional to its log-variance.

Now observe that the unsupervised Deep SVDD objective (3) (disregarding weight decay regularization) is equivalent to minimizing the empirical variance thus minimizing an approximate upper bound for the entropy of the latent distribution.

Since the Deep SVDD network is pre-trained on an autoencoding objective [56] that implicitly maximizes the mutual information $\mathcal{I}(X; Z)$, we can interpret Deep SVDD as following the Infomax principle (2) with the additional objective that the latent distribution should have low entropy.

3.3 Deep SAD

We are happy to now introduce our *Deep SAD* method. Assume that, in addition to the n unlabeled samples $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \in \mathcal{X}$ with $\mathcal{X} \subseteq \mathbb{R}^D$, we have access to m labeled samples $(\tilde{\boldsymbol{x}}_1, \tilde{y}_1), \ldots, (\tilde{\boldsymbol{x}}_m, \tilde{y}_m) \in \mathcal{X} \times \mathcal{Y}$ and $\mathcal{Y} = \{-1, +1\}$. We denote $\tilde{y} = +1$ for known normal examples and $\tilde{y} = -1$ for known anomalies.

Following the insights above, we formulate our deep semi-supervised AD objective under the idea that the latent distribution of the normal data, $Z^+ = Z|\{Y{=}{+1}\}$, should have low entropy, whereas the latent distribution of anomalies, $Z^- = Z|\{Y{=}{-1}\}$, should have high entropy. By this, we do not impose any additional assumption on the anomaly-generating distribution $X|\{Y{=}{-1}\}$, such as a manifold or cluster assumption that supervised or semi-supervised classification approaches commonly make [75, 11]. We argue that such a model better captures the nature of anomalies, which can be thought of as being generated from an infinite mixture of all distributions that are different from the normal data distribution, indubitably a distribution that has high entropy. We can express this idea in terms of principle (2) with respective entropy regularization of the latent distribution:

$$\max_{p(z|x)} \quad \mathcal{I}(X; Z) + \beta \, (\mathcal{H}(Z^-) - \mathcal{H}(Z^+)). \tag{8}$$

Based on the connection between Deep SVDD and entropy minimization we have shown in Section 3.2, we define our *Deep SAD* objective as

$$\min_{\mathcal{W}} \quad \frac{1}{n+m}\sum_{i=1}^{n}\|\phi(\boldsymbol{x}_i;\mathcal{W})-\boldsymbol{c}\|^2+\frac{\eta}{n+m}\sum_{j=1}^{m}\left(\|\phi(\tilde{\boldsymbol{x}}_j;\mathcal{W})-\boldsymbol{c}\|^2\right)^{\tilde{y}_j}+\frac{\lambda}{2}\sum_{\ell=1}^{L}\|\boldsymbol{W}^{\ell}\|_F^2, \qquad (9)$$

with hyperparameters $\eta > 0$ and $\lambda > 0$. We again impose a quadratic loss on the distances of the mapped points to the fixed center $\boldsymbol{c}$, for both the unlabeled as well as the labeled normal examples, thus intending to learn a latent distribution with low entropy for the normal data. This also incorporates the assumption common in AD that most of the unlabeled data is normal. In contrast, for the labeled anomalies we penalize the *inverse* of the distances such that anomalies must be mapped further away from the center.[1] That is, we penalize low variance and thus the network must attempt to map known anomalies to a heavy-tailed distribution that has high entropy. To maximize the mutual information $\mathcal{I}(X;Z)$ in (8), we also rely on autoencoder pre-training.

The hyperparameter $\eta > 0$ controls the balance between the labeled and unlabeled terms, where $\eta < 1$ emphasizes the unlabeled and $\eta > 1$ the labeled objective. For $\eta = 1$, the two terms are weighted equally. The last term is a weight decay regularizer. Note that we recover the unsupervised Deep SVDD (3) formulation as the special case where only unlabeled data is available ($m = 0$). As an anomaly score, we again take the distance of the latent representation to the center c as in (4). We optimize the generally non-convex Deep SAD objective (9) via SGD using backpropagation. Appendix A in the supplementary material provides further details.

4 Experiments

We evaluate Deep SAD on MNIST, Fashion-MNIST, and CIFAR-10 as well as classic anomaly detection benchmark datasets. We compare to shallow, hybrid, as well as deep unsupervised, semi-supervised and supervised competitors. We refer to other recent works [56, 22, 25] for further comprehensive comparisons solely between unsupervised deep AD methods.[2]

4.1 Competing Methods

We consider the OC-SVM [60] and SVDD [65] with Gaussian kernel (which are in this case equivalent), Isolation Forest [38], and KDE [50] as shallow unsupervised baselines. For unsupervised deep competitors, we consider the well-established autoencoder and the state-of-the-art unsupervised Deep SVDD method [56]. For semi-supervised approaches, we consider the shallow state-of-the-art semi-supervised AD method of SSAD [23] with Gaussian kernel. As mentioned previously, there are no deep methods for semi-supervised AD that are applicable to the general multivariate data setting. However, we add the well-known Semi-Supervised Deep Generative Model (SS-DGM) [34] to make a comparison with a deep semi-supervised classifier. To complete the full learning spectrum, we also compare to a fully supervised deep classifier trained on the binary cross-entropy loss. Finally, in addition to training the shallow detectors on the raw input features, we also consider all their hybrid variants of applying them to the bottleneck representation given by the autoencoder [19, 47].

In our experiments we deliberately grant the shallow and hybrid methods an unfair advantage by selecting their hyperparameters to maximize AUC on a subset (10%) of the test set to establish strong baselines. To control for architectural effects between the competing deep methods, we always employ the same (LeNet-type) deep networks. Full details on network architectures and hyperparameter selection can be found in Appendices B and C of the supplementary material. Due to space constraints, in the main text we only report results for methods which showed competitive performance and defer results for the under-performing methods in Appendix D.

4.2 Experimental Scenarios on MNIST, Fashion-MNIST, and CIFAR-10

Semi-supervised anomaly detection setup The MNIST, Fashion-MNIST, and CIFAR-10 datasets all have ten classes from which we derive ten AD setups on each dataset. In every setup, we set one of the ten classes to be the normal class and let the remaining nine classes represent anomalies. We use

[1] To ensure numerical stability, we add a machine epsilon (`eps` $\sim 10^{-6}$) to the denominator of the inverse.

[2] Our code is available at: `https://github.com/lukasruff/Deep-SAD-PyTorch`

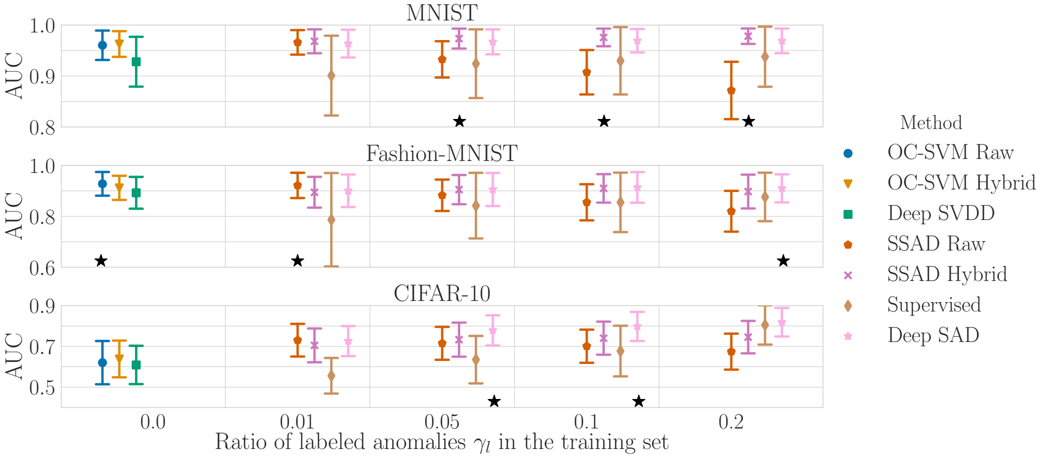


Figure 2: Results of experimental scenario (i), where we increase the ratio of labeled anomalies γ_l in the training set. We report the avg. AUC with st. dev. computed over 90 experiments at various ratios γ_l. A "$\star$" indicates a statistically significant ($\alpha = 0.05$) difference between 1st and 2nd.

the original training data of the respective normal class as the unlabeled part of our training set. Thus we start with a clean anomaly detection setting that fulfills the assumption that most (in this case all) unlabeled samples are normal. The training data of the respective nine anomaly classes then forms the data pool from which we draw anomalies for training to create different scenarios. We compute the AUC metric on the original respective test sets using ground truth labels to make a quantitative comparison, i.e. $\tilde{y} = +1$ for the normal class and $\tilde{y} = -1$ for the respective nine anomaly classes. We rescale pixels to $[0, 1]$ via min-max feature scaling as the only data pre-processing step.

Experimental scenarios We examine three scenarios in which we vary the following three experimental parameters: (i) the ratio of labeled training data γ_l, (ii) the ratio of pollution γ_p in the unlabeled training data with (unknown) anomalies, and (iii) the number of anomaly classes k_l included in the labeled training data.

(i) Adding labeled anomalies In this scenario, we investigate the effect that including labeled anomalies into training has on detection performance and potential advantage of using a semi-supervised AD method over other paradigms. To do this we increase the ratio of labeled training data $\gamma_l = m/(n+m)$ adding more and more known anomalies $\tilde{\boldsymbol{x}}_1, \ldots, \tilde{\boldsymbol{x}}_m$ with $\tilde{y}_j = -1$ to the training set. We add the labeled anomalies from $k_l = 1$ anomaly class (out of the nine remaining ones). For testing, we then consider all nine remaining classes as anomalies, i.e. there are eight novel classes at testing time. We do this to simulate the unpredictable nature of anomalies. For the unlabeled part of the training set, we keep the training data of the respective normal class, which we leave unpolluted for now, i.e. $\gamma_p = 0$. We iterate this training set generation process per AD setup always over all the nine respective anomaly classes and report the average results over the ten AD setups × nine anomaly classes, i.e. over 90 experiments per labeled ratio γ_l.

(ii) Polluted training data Here we investigate the robustness of the different methods to an increasing pollution ratio γ_p of the training set with unknown anomalies. To do so we pollute the unlabeled part of the training set with anomalies drawn from all nine respective anomaly classes in each AD setup. We fix the ratio of labeled training samples at $\gamma_l = 0.05$ where we again draw samples only from $k_l = 1$ anomaly class in this scenario. We repeat this training set generation process per AD setup over all the nine respective anomaly classes and report the average results over the resulting 90 experiments per pollution ratio γ_p. We hypothesize that the semi-supervised approach alleviates the negative impact pollution has on detection performance, since labeled anomalies should help to "filter out" similar unknown anomalies.

(iii) Number of known anomaly classes In the last scenario, we compare the detection performance at various numbers of known anomaly classes. In scenarios (i) and (ii), we have always sampled labeled anomalies only from $k_l = 1$ out of the nine anomaly classes. In this scenario, we now increase the number of anomaly classes k_l included in the labeled part of the training set. Since we have a limited number of anomaly classes (nine) in each AD setup, we expect the supervised

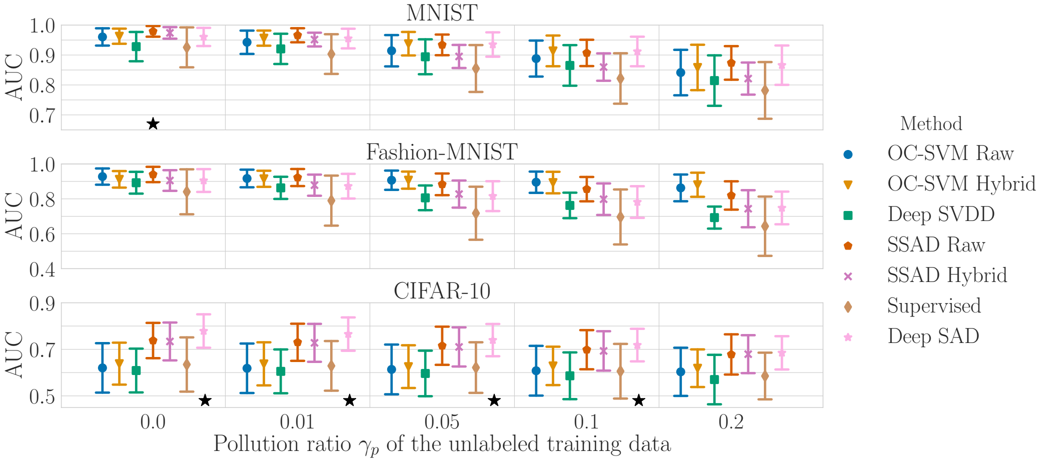


Figure 3: Results of experimental scenario (ii), where we pollute the unlabeled part of the training set with (unknown) anomalies. We report the avg. AUC with st. dev. computed over 90 experiments at various ratios γ_p. A "$\star$" indicates a statistically significant ($\alpha = 0.05$) difference between 1st and 2nd.

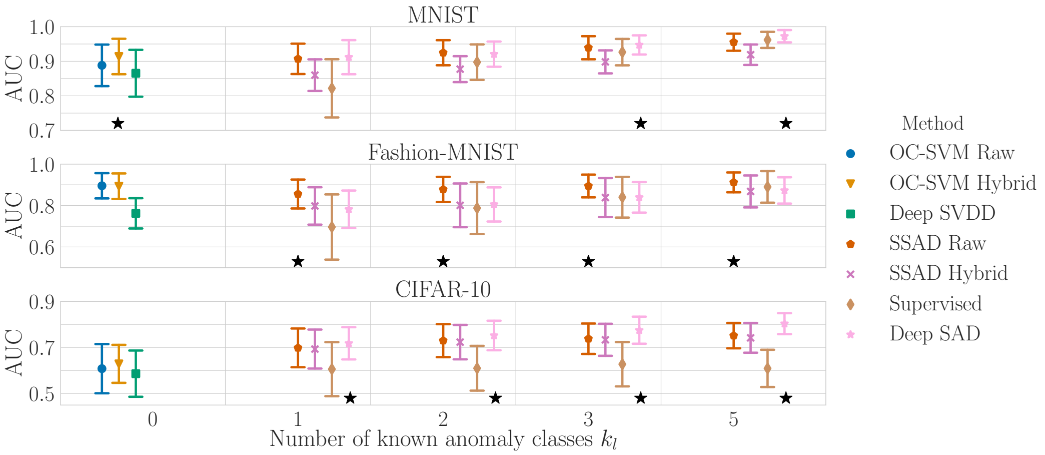


Figure 4: Results of experimental scenario (iii), where we increase the number of anomaly classes k_l included in the labeled training data. We report the avg. AUC with st. dev. computed over 100 experiments at various numbers k_l. A "$\star$" indicates a statistically significant ($\alpha = 0.05$) difference between 1st and 2nd.

classifier to catch up at some point. We fix the overall ratio of labeled training examples again at $\gamma_l = 0.05$ and consider a pollution ratio of $\gamma_p = 0.1$ for the unlabeled training data in this scenario. We repeat this training set generation process for ten seeds in every of the ten AD setups and report the average results over the resulting 100 experiments per number k_l. For every seed, the k_l classes are drawn uniformly at random out of the nine respective anomaly classes.

Results The results of the scenarios (i)–(iii) are shown in Figures 2–4. In addition to reporting the average AUC with standard deviation, we always conduct Wilcoxon signed-rank tests [71] between the best and second best performing method and indicate statistically significant ($\alpha = 0.05$) differences. Figure 2 demonstrates the advantage of a semi-supervised approach to AD especially on the most complex CIFAR-10 dataset, where Deep SAD performs best. Moreover, Figure 2 confirms that a supervised approach is vulnerable to novel anomalies at testing when only little labeled training data is available. In comparison, our Deep SAD generalizes to novel anomalies while also taking advantage of the labeled examples. Note that the hybrid SSAD, which has not yet been considered in the literature, also proves to be a sound baseline. Figure 3 shows that the detection performance of all methods decreases with increasing data pollution. Deep SAD proves to be most robust again especially on the most complex CIFAR-10. Finally, Figure 4 shows that the more diverse the labeled anomalies in the training set are, the better the detection performance becomes. We also see that the

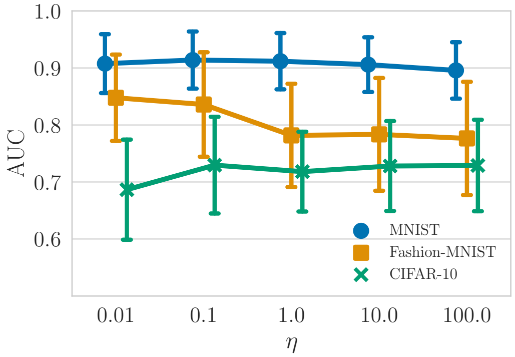


Figure 5: Deep SAD sensitivity analysis w.r.t. η. We report the avg. AUC with st. dev. computed over 90 experiments for various values of η.

Table 1: Anomaly detection benchmarks.

Dataset	N	D	#outliers (%)
arrhythmia	452	274	66 (14.6%)
cardio	1,831	21	176 (9.6%)
satellite	6,435	36	2,036 (31.6%)
satimage-2	5,803	36	71 (1.2%)
shuttle	49,097	9	3,511 (7.2%)
thyroid	3,772	6	93 (2.5%)

supervised method is very sensitive to the number of attack classes but catches up at some point as suspected. Overall, we observe that Deep SAD is particularly advantageous on complex data.

Hyperparameter η sensitivity analysis We run Deep SAD experiments on the ten AD setups from above on each dataset for $\eta \in \{10^{-2}, \dots, 10^{2}\}$ to analyze the sensitivity of Deep SAD with respect to the hyperparameter $\eta > 0$. In this analysis, we set the experimental parameters to $\gamma_l = 0.05$, $\gamma_p = 0.1$, and $k_l = 1$ and again iterate over all nine anomaly classes in every AD setup. The results shown in Figure 5 suggest that Deep SAD is fairly robust against changes of the hyperparameter η.

Table 2: Results on classic AD benchmark datasets in the setting with no pollution $\gamma_p = 0$ and a ratio of labeled anomalies of $\gamma_l = 0.01$ in the training set. We report the avg. AUC with st. dev. computed over 10 seeds. A "$\star$" indicates a statistically significant ($\alpha = 0.05$) difference between 1st and 2nd.

	OC-SVM	**OC-SVM**	**Deep**	**SSAD**	**SSAD**	**Supervised**	**Deep**
Dataset	**Raw**	**Hybrid**	**SVDD**	**Raw**	**Hybrid**	**Classifier**	**SAD**
arrhythmia	84.5±3.9	76.7±6.2	74.6±9.0	**86.7±4.0***	78.3±5.1	39.2±9.5	75.9±8.7
cardio	98.5±0.3	82.8±9.3	84.8±3.6	**98.8±0.3**	86.3±5.8	83.2±9.6	95.0±1.6
satellite	95.1±0.2	68.6±4.8	79.8±4.1	**96.2±0.3***	86.9±2.8	87.2±2.1	91.5±1.1
satimage-2	99.4±0.8	96.7±2.1	98.3±1.4	**99.9±0.1**	96.8±2.1	**99.9±0.1**	**99.9±0.1**
shuttle	99.4±0.9	94.1±9.5	86.3±7.5	**99.6±0.5**	97.7±1.0	95.1±8.0	98.4±0.9
thyroid	98.3±0.9	91.2±4.0	72.0±9.7	97.9±1.9	95.3±3.1	97.8±2.6	**98.6±0.9**

4.3 Classic Anomaly Detection Benchmark Datasets

In the last experiment, we examine the detection performance on some well-established AD benchmark datasets [54] listed in Table 1. We do this to evaluate the deep against the shallow approaches also on non-image, tabular datasets that are rarely considered in the deep AD literature. For the evaluation, we consider random train-to-test set splits of 60:40 while maintaining the original proportion of anomalies in each set. We then run experiments for 10 seeds with $\gamma_l = 0.01$ and $\gamma_p = 0$, i.e. 1% of the training set are labeled anomalies and the unlabeled training data is unpolluted. Since there are no specific different anomaly classes, we set $k_l = 1$. We standardize features to have zero mean and unit variance as the only pre-processing step.

Table 2 shows the results. We observe that the shallow kernel methods seem to perform slightly better on the rather small, low-dimensional benchmarks. Deep SAD proves competitive though and the small differences might be explained by the strong advantage we deliberately grant the shallow methods in the selection of their hyperparameters. The results in section 4.2 and other recent works [56, 22, 25] demonstrate that deep methods are especially superior on complex data with hierarchical structure. Unlike other deep approaches [20, 35, 42, 17, 22], however, our Deep SAD method is not domain or data-type specific. Due to its strong performance using both deep and shallow networks we expect Deep SAD to extend well to other data types.

5 Conclusion

We have introduced Deep SAD, a deep method for semi-supervised anomaly detection. To derive our method, we formulated an information-theoretic perspective on deep anomaly detection. Our experiments demonstrate that Deep SAD improves detection performance especially on more complex datasets already with only small amounts of labeled data. Our results suggest that semi-supervised approaches to anomaly detection should always be preferred in applications whenever some labeled information is available.

Acknowledgments

LR acknowledges support from the German Ministry of Education and Research (BMBF) in the project ALICE III (FKZ: 01IS18049B). MK and RV acknowledge support by the German Research Foundation (DFG) award KL 2698/2-1 and by the German Ministry of Education and Research (BMBF) awards 031L0023A, 01IS18051A, and 031B0770E. Part of the work was done while MK was a sabbatical visitor of the DASH Center at the University of Southern California. AB is grateful for support by the Singapore Ministry of Education grant MOE2016-T2-2-154. This work was supported by the German Ministry for Education and Research (BMBF) as Berlin Big Data Center (01IS14013A) and Berlin Center for Machine Learning (01IS18037I). Partial funding by DFG is acknowledged (EXC 2046/1, project-ID: 390685689). This work was also supported by the Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (No. 2017-0-00451, No. 2017-0-01779).

References

[1] A. Achille and S. Soatto. Emergence of invariance and disentanglement in deep representations. *Journal of Machine Learning Research*, 19(1):1947–1980, 2018.

[2] A. Alemi, I. Fischer, J. V. Dillon, and K. Murphy. Deep variational information bottleneck. In *International Conference on Learning Representations*, 2017.

[3] A. Alemi, B. Poole, I. Fischer, J. Dillon, R. A. Saurous, and K. Murphy. Fixing a broken ELBO. In *International Conference on Machine Learning*, volume 80, pages 159–168, 2018.

[4] J. T. A. Andrews, E. J. Morton, and L. D. Griffin. Detecting Anomalous Data Using Auto-Encoders. *International Journal of Machine Learning and Computing*, 6(1):21, 2016.

[5] S. Arora, N. Cohen, and E. Hazan. On the optimization of deep networks: Implicit acceleration by overparameterization. In *International Conference on Machine Learning*, volume 80, pages 244–253, 2018.

[6] M. Belkin, S. Ma, and S. Mandal. To understand deep learning we need to understand kernel learning. In *International Conference on Machine Learning*, pages 540–548, 2018.

[7] A. J. Bell and T. J. Sejnowski. An information-maximization approach to blind separation and blind deconvolution. *Neural Computation*, 7(6):1129–1159, 1995.

[8] G. Blanchard, G. Lee, and C. Scott. Semi-supervised novelty detection. *Journal of Machine Learning Research*, 11(Nov):2973–3009, 2010.

[9] R. Chalapathy and S. Chawla. Deep learning for anomaly detection: A survey. *arXiv preprint arXiv:1901.03407*, 2019.

[10] V. Chandola, A. Banerjee, and V. Kumar. Anomaly Detection: A Survey. *ACM Computing Surveys*, 41(3): 1–58, 2009.

[11] O. Chapelle, B. Schölkopf, and A. Zien. Semi-supervised learning. *IEEE Transactions on Neural Networks*, 20(3):542–542, 2009.

[12] J. Chen, S. Sathe, C. Aggarwal, and D. Turaga. Outlier detection with autoencoder ensembles. In *Proceedings of the 2017 SIAM International Conference on Data Mining*, pages 90–98, 2017.

[13] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel. InfoGAN: Interpretable representation learning by information maximizing generative adversarial nets. In *Advances in Neural Information Processing Systems*, pages 2172–2180, 2016.

[14] N. Cohen, O. Sharir, and A. Shashua. On the expressive power of deep learning: A tensor analysis. In *International Conference on Algorithmic Learning Theory*, volume 49, pages 698–728, 2016.

[15] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. John Wiley & Sons, 2012.

[16] Z. Dai, Z. Yang, F. Yang, W. W. Cohen, and R. R. Salakhutdinov. Good semi-supervised learning that requires a bad gan. In *Advances in Neural Information Processing Systems*, pages 6510–6520, 2017.

[17] L. Deecke, R. A. Vandermeulen, L. Ruff, S. Mandt, and M. Kloft. Image anomaly detection with generative adversarial networks. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 3–17, 2018.

[18] R. Eldan and O. Shamir. The power of depth for feedforward neural networks. In *International Conference on Algorithmic Learning Theory*, volume 49, pages 907–940, 2016.

[19] S. M. Erfani, S. Rajasegarar, S. Karunasekera, and C. Leckie. High-dimensional and large-scale anomaly detection using a linear one-class SVM with deep learning. *Pattern Recognition*, 58:121–134, 2016.

[20] T. Ergen, A. H. Mirza, and S. S. Kozat. Unsupervised and semi-supervised anomaly detection with LSTM neural networks. *arXiv:1710.09207*, 2017.

[21] X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In *International Conference on Artificial Intelligence and Statistics*, pages 249–256, 2010.

[22] I. Golan and R. El-Yaniv. Deep anomaly detection using geometric transformations. In *Advances in Neural Information Processing Systems*, pages 9758–9769, 2018.

[23] N. Görnitz, M. Kloft, K. Rieck, and U. Brefeld. Toward supervised anomaly detection. *Journal of Artificial Intelligence Research*, 46:235–262, 2013.

[24] S. Hawkins, H. He, G. Williams, and R. Baxter. Outlier Detection Using Replicator Neural Networks. In *International Conference on Data Warehousing and Knowledge Discovery*, volume 2454, pages 170–180, 2002.

[25] D. Hendrycks, M. Mazeika, and T. G. Dietterich. Deep anomaly detection with outlier exposure. In *International Conference on Learning Representations*, 2019.

[26] G. E. Hinton and R. R. Salakhutdinov. Reducing the Dimensionality of Data with Neural Networks. *Science*, 313(5786):504–507, 2006.

[27] R. D. Hjelm, A. Fedorov, S. Lavoie-Marchildon, K. Grewal, A. Trischler, and Y. Bengio. Learning deep representations by mutual information estimation and maximization. In *International Conference on Learning Representations*, 2019.

[28] M. D. Hoffman and M. J. Johnson. ELBO surgery: yet another way to carve up the variational evidence lower bound. In *NIPS Workshop in Advances in Approximate Bayesian Inference*, 2016.

[29] S. Ioffe and C. Szegedy. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In *International Conference on Machine Learning*, pages 448–456, 2015.

[30] X. Ji, J. F. Henriques, and A. Vedaldi. Invariant information distillation for unsupervised image segmentation and clustering. *arXiv preprint arXiv:1807.06653*, 2018.

[31] J. Kim and C. D. Scott. Robust kernel density estimation. *Journal of Machine Learning Research*, 13(Sep): 2529–2565, 2012.

[32] D. P. Kingma and J. Ba. Adam: A Method for Stochastic Optimization. *arXiv:1412.6980*, 2014.

[33] D. P. Kingma and M. Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[34] D. P. Kingma, S. Mohamed, D. J. Rezende, and M. Welling. Semi-supervised learning with deep generative models. In *Advances in Neural Information Processing Systems*, pages 3581–3589, 2014.

[35] B. Kiran, D. Thomas, and R. Parakkal. An overview of deep learning based methods for unsupervised and semi-supervised anomaly detection in videos. *Journal of Imaging*, 4(2):36, 2018.

[36] S. Lapuschkin, S. Wäldchen, A. Binder, G. Montavon, W. Samek, and K.-R. Müller. Unmasking clever hans predictors and assessing what machines really learn. *Nature Communications*, 10(1):1096, 2019.

[37] R. Linsker. Self-organization in a perceptual network. *IEEE Computer*, 21(3):105–117, 1988.

[38] F. T. Liu, K. M. Ting, and Z.-H. Zhou. Isolation Forest. In *International Conference on Data Mining*, pages 413–422, 2008.

[39] Y. Liu and Y. F. Zheng. Minimum enclosing and maximum excluding machine for pattern description and discrimination. In *International Conference on Pattern Recognition*, volume 3, pages 129–132, 2006.

[40] A. Makhzani and B. Frey. K-sparse autoencoders. In *International Conference on Learning Representations*, 2014.

[41] A. Makhzani, J. Shlens, N. Jaitly, I. Goodfellow, and B. Frey. Adversarial autoencoders. In *International Conference on Learning Representations*, 2015.

[42] E. Min, J. Long, Q. Liu, J. Cui, Z. Cai, and J. Ma. SU-IDS: A semi-supervised and unsupervised framework for network intrusion detection. In *International Conference on Cloud Computing and Security*, pages 322–334, 2018.

[43] G. Montavon, M. L. Braun, and K.-R. Müller. Kernel analysis of deep networks. *Journal of Machine Learning Research*, 12(Sep):2563–2581, 2011.

[44] M. M. Moya, M. W. Koch, and L. D. Hostetler. One-class classifier networks for target recognition applications. In *Proceedings World Congress on Neural Networks*, pages 797–801, 1993.

[45] J. Muñoz-Marí, F. Bovolo, L. Gómez-Chova, L. Bruzzone, and G. Camp-Valls. Semi-Supervised One-Class Support Vector Machines for Classification of Remote Sensing Sata. *IEEE Transactions on Geoscience and Remote Sensing*, 48(8):3188–3197, 2010.

[46] B. Neyshabur, S. Bhojanapalli, D. McAllester, and N. Srebro. Exploring generalization in deep learning. In *Advances in Neural Information Processing Systems*, pages 5947–5956, 2017.

[47] M. Nicolau, J. McDermott, et al. A hybrid autoencoder and density estimation model for anomaly detection. In *International Conference on Parallel Problem Solving from Nature*, pages 717–726, 2016.

[48] A. Odena. Semi-supervised learning with generative adversarial networks. *arXiv:1606.01583*, 2016.

[49] A. Oliver, A. Odena, C. A. Raffel, E. D. Cubuk, and I. Goodfellow. Realistic evaluation of deep semi-supervised learning algorithms. In *Advances in Neural Information Processing Systems*, pages 3235–3246, 2018.

[50] E. Parzen. On Estimation of a Probability Density Function and Mode. *The Annals of Mathematical Statistics*, 33(3):1065–1076, 1962.

[51] M. A. Pimentel, D. A. Clifton, L. Clifton, and L. Tarassenko. A review of novelty detection. *Signal Processing*, 99:215–249, 2014.

[52] M. Raghu, B. Poole, J. Kleinberg, S. Ganguli, and J. S. Dickstein. On the expressive power of deep neural networks. In *International Conference on Machine Learning*, volume 70, pages 2847–2854, 2017.

[53] A. Rasmus, M. Berglund, M. Honkala, H. Valpola, and T. Raiko. Semi-supervised learning with ladder networks. In *Advances in Neural Information Processing Systems*, pages 3546–3554, 2015.

[54] S. Rayana. ODDS library, 2016. URL `http://odds.cs.stonybrook.edu`.

[55] D. J. Rezende, S. Mohamed, and D. Wierstra. Stochastic Backpropagation and Approximate Inference in Deep Generative Models. In *International Conference on Machine Learning*, volume 32, pages 1278–1286, 2014.

[56] L. Ruff, R. A. Vandermeulen, N. Görnitz, L. Deecke, S. A. Siddiqui, A. Binder, E. Müller, and M. Kloft. Deep one-class classification. In *International Conference on Machine Learning*, volume 80, pages 4390–4399, 2018.

[57] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning internal representations by error propagation. In *Parallel Distributed Processing – Explorations in the Microstructure of Cognition*, chapter 8, pages 318–362. MIT Press, 1986.

[58] M. Sakurada and T. Yairi. Anomaly detection using autoencoders with nonlinear dimensionality reduction. In *Proceedings of the 2nd MLSDA Workshop*, page 4, 2014.

[59] A. M. Saxe, Y. Bansal, J. Dapello, M. Advani, A. Kolchinsky, B. D. Tracey, and D. D. Cox. On the information bottleneck theory of deep learning. In *International Conference on Learning Representations*, 2018.

[60] B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson. Estimating the Support of a High-Dimensional Distribution. *Neural Computation*, 13(7):1443–1471, 2001.

[61] C. D. Scott and R. D. Nowak. Learning minimum volume sets. *Journal of Machine Learning Research*, 7 (Apr):665–704, 2006.

[62] C. E. Shannon. A mathematical theory of communication. *Bell system technical journal*, 27(3):379–423, 1948.

[63] R. Shwartz-Ziv and N. Tishby. Opening the black box of deep neural networks via information. *arXiv preprint arXiv:1703.00810*, 2017.

[64] N. Slonim, G. S. Atwal, G. Tkačik, and W. Bialek. Information-based clustering. *Proceedings of the National Academy of Sciences*, 102(51):18297–18302, 2005.

[65] D. M. J. Tax and R. P. W. Duin. Support Vector Data Description. *Machine Learning*, 54(1):45–66, 2004.

[66] N. Tishby and N. Zaslavsky. Deep learning and the information bottleneck principle. In *IEEE Information Theory Workshop*, pages 1–5, 2015.

[67] N. Tishby, F. C. Pereira, and W. Bialek. The information bottleneck method. In *The 37th annual Allerton Conference on Communication, Control and Computing*, pages 368–377, 1999.

[68] R. Vandermeulen and C. Scott. Consistency of robust kernel density estimators. In *Conference on Learning Theory*, pages 568–591, 2013.

[69] J. Wang, P. Neskovic, and L. N. Cooper. Pattern classification via single spheres. In *International Conference on Discovery Science*, pages 241–252. Springer, 2005.

[70] T. Wiatowski and H. Bölcskei. A mathematical theory of deep convolutional neural networks for feature extraction. *IEEE Transactions on Information Theory*, 64(3):1845–1866, 2018.

[71] F. Wilcoxon. Individual comparisons by ranking methods. *Biometrics Bulletin*, 1(6):80–83, 1945.

[72] S. Zhai, Y. Cheng, W. Lu, and Z. Zhang. Deep structured energy based models for anomaly detection. In *International Conference on Machine Learning*, volume 48, pages 1100–1109, 2016.

[73] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals. Understanding deep learning requires rethinking generalization. In *International Conference on Learning Representations*, 2017.

[74] S. Zhao, J. Song, and S. Ermon. InfoVAE: Information maximizing variational autoencoders. *arXiv preprint arXiv:1706.02262*, 2017.

[75] X. Zhu. Semi-supervised learning literature survey. *Computer Sciences TR 1530, University of Wisconsin Madison*, 2008.

A Optimization of Deep SAD

The Deep SAD objective is generally non-convex in the network weights $\mathcal{W}$ which usually is the case in deep learning. For a computationally efficient optimization, we rely on (mini-batch) SGD to optimize the network weights using the backpropagation algorithm. For improved generalization, we add ℓ_2 weight decay regularization with hyperparameter $\lambda > 0$ to the objective. Algorithm 1 summarizes the Deep SAD optimization routine.

Algorithm 1 Optimization of Deep SAD

```
Input:
    Unlabeled data: $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$
    Labeled data: $(\boldsymbol{x}'_1, y'_1), \ldots, (\boldsymbol{x}'_m, y'_m)$
    Hyperparameters: $\eta, \lambda$
    SGD learning rate: $\varepsilon$
Output:
    Trained model: $\mathcal{W}^*$

Initialize:
        Neural network weights: $\mathcal{W}$
        Hypersphere center: $\boldsymbol{c}$
for each epoch do
    for each mini-batch do
        Draw mini-batch $\mathcal{B}$
        $\mathcal{W} \leftarrow \mathcal{W} - \varepsilon \cdot \nabla_{\mathcal{W}} J(\mathcal{W}; \mathcal{B})$
    end for
end for
```

Using SGD allows Deep SAD to scale with large datasets as the computational complexity scales linearly in the number of training batches and computations in each batch can be parallelized (e.g., by training on GPUs). Moreover, Deep SAD has low memory complexity as a trained model is fully characterized by the final network parameters $\mathcal{W}^*$ and no data must be saved or referenced for prediction. Instead, the prediction only requires a forward pass on the network which usually is just a concatenation of simple functions. This enables fast predictions for Deep SAD.

Initialization of the network weights $\mathcal{W}$ We establish an autoencoder pre-training routine for initialization. That is, we first train an autoencoder that has an encoder with the same architecture as network ϕ on the reconstruction loss (mean squared error or cross-entropy loss). After training, we then initialize $\mathcal{W}$ with the converged parameters of the encoder. Note that this is in line with the Infomax principle (2) for unsupervised representation learning.

Initialization of the center $\boldsymbol{c}$ After initializing the network weights $\mathcal{W}$, we fix hypersphere center $\boldsymbol{c}$ as the mean of the network representations that we obtain from an initial forward pass on the data (excluding labeled anomalies). We found SGD convergence to be smoother and faster by fixing center $\boldsymbol{c}$ in the neighborhood of the initial data representations as we already observed in Ruff et al. [56]. If some labeled normal examples are available, using only those examples for a mean initialization would be another strategy to minimize possible distortions from polluted unlabeled training data. Adding center $\boldsymbol{c}$ to the optimization variables would allow a trivial "hypersphere collapse" solution for unsupervised Deep SVDD.

Preventing a hypersphere collapse A "hypersphere collapse" describes the trivial solution that neural network ϕ converges to the constant function $\phi \equiv \boldsymbol{c}$, i.e. the hypersphere collapses to a single point. In Ruff et al. [56], we demonstrate theoretical network properties that prevent such a collapse which we adopt for Deep SAD. Most importantly, network ϕ must have no bias terms and no bounded activation functions. We refer to Ruff et al. [56] for further details. If there are sufficiently many labeled anomalies available for training, however, hypersphere collapse is not a problem for Deep SAD due to the opposing labeled and unlabeled objectives.

B Network Architectures

We employ LeNet-type convolutional neural networks (CNNs) on MNIST, Fashion-MNIST, and CIFAR-10, where each convolutional module consists of a convolutional layer followed by leaky ReLU activations with leakiness $\alpha = 0.1$ and (2×2)-max-pooling. On MNIST, we employ a CNN with two modules, $8\times(5\times5)$-filters followed by $4\times(5\times5)$-filters, and a final dense layer of 32 units. On Fashion-MNIST, we employ a CNN also with two modules, $16\times(5\times5)$-filters and $32\times(5\times5)$-filters, followed by two dense layers of 64 and 32 units respectively. On CIFAR-10, we employ a CNN with three modules, $32\times(5\times5)$-filters, $64\times(5\times5)$-filters, and $128\times(5\times5)$-filters, followed by a final dense layer of 128 units.

On the classic AD benchmark datasets, we employ standard MLP feed-forward architectures. On arrhythmia, a 3-layer MLP with 128-64-32 units. On cardio, satellite, satimage-2, and shuttle a 3-layer MLP with 32-16-8 units. On thyroid a 3-layer MLP with 32-16-4 units.

C Details on Competing Methods

OC-SVM/SVDD The OC-SVM and SVDD are equivalent for the Gaussian/RBF kernel we employ. As mentioned in the main paper, we deliberately grant the OC-SVM/SVDD an unfair advantage by selecting its hyperparameters to maximize AUC on a subset (10%) of the test set to establish a strong baseline. To do this, we consider the RBF scale parameter $\gamma \in \{2^{-7}, 2^{-6}, \ldots 2^{2}\}$ and select the best performing one. Moreover, we always repeat this over ν-parameter $\nu \in \{0.01, 0.05, 0.1, 0.2, 0.5\}$ and then report the best final result.

Isolation Forest (IF) We set the number of trees to $t = 100$ and the sub-sampling size to $\psi = 256$, as recommended in the original work [38].

Kernel Density Estimator (KDE) We select the bandwidth h of the Gaussian kernel from $h \in \{2^{0.5}, 2^{1}, \ldots, 2^{5}\}$ via 5-fold cross-validation using the log-likelihood score following [56].

SSAD We also deliberately grant the state-of-the-art semi-supervised AD kernel method SSAD the unfair advantage of selecting its hyperparameters optimally to maximize AUC on a subset (10%) of the test set. To do this, we again select the scale parameter γ of the RBF kernel we use from $\gamma \in \{2^{-7}, 2^{-6}, \ldots 2^{2}\}$ and select the best performing one. Otherwise we set the hyperparameters as recommend by the original authors to $\kappa = 1$, $\kappa = 1$, $\eta_u = 1$, and $\eta_l = 1$ [23].

(Convolutional) Autoencoder ((C)AE) To create the (convolutional) autoencoders, we symmetrically construct the decoders w.r.t. the architectures reported in Appenidx B, which make up the encoder parts of the autoencoders. Here, we replace max-pooling with simple upsampling and convolutions with deconvolutions. We train the autoencoders on the MSE reconstruction loss that also serves as the anomaly score.

Hybrid Variants To establish hybrid methods, we apply the OC-SVM, IF, KDE, and SSAD as outlined above to the resulting bottleneck representations given by the converged autoencoder.

Unsupervised Deep SVDD We consider both variants, Soft-Boundary Deep SVDD and One-Class Deep SVDD as unsupervised baselines and always report the better performance as the unsupervised result. For Soft-Boundary Deep SVDD, we optimally solve for the radius R on every mini-batch and run experiments for $\nu \in \{0.01, 0.1\}$. We set the weight decay hyperparameter to $\lambda = 10^{-6}$. For Deep SVDD, we remove all bias terms from the network to prevent a hypersphere collapse as we recommended in the original work [56].

Deep SAD We set $\lambda = 10^{-6}$ and equally weight the unlabeled and labeled examples by setting $\eta = 1$ if not reported otherwise.

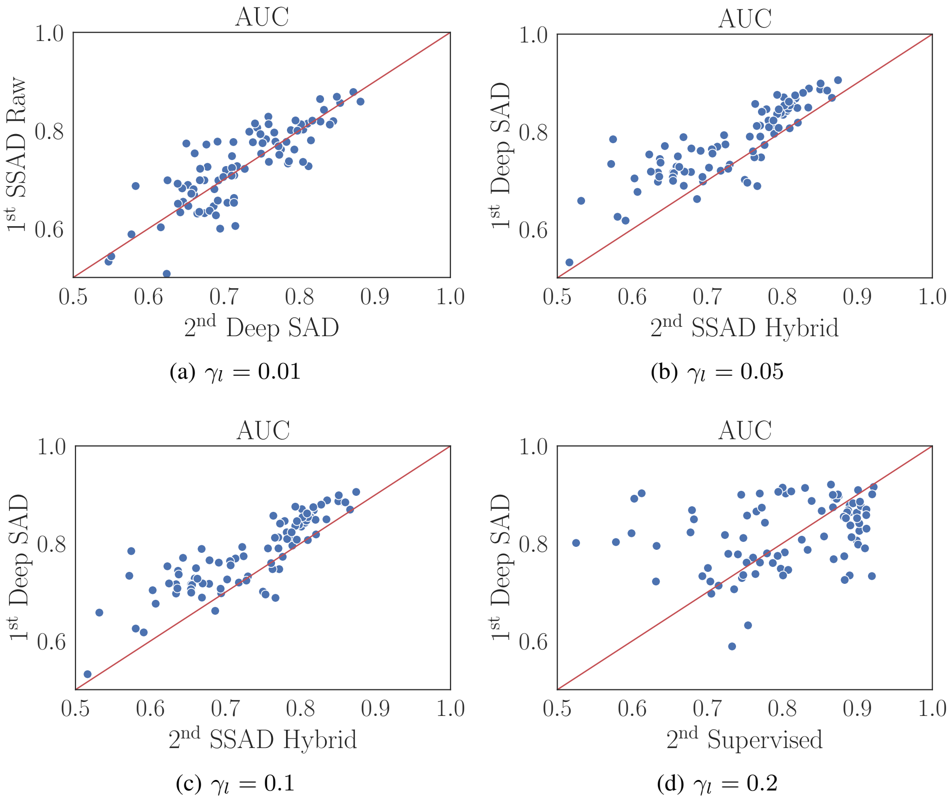
SS-DGM We consider both the M2 and M1+M2 model and always report the better performing result. Otherwise we follow the settings as recommended in the original work [34].

Supervised Deep Binary Classifier To interpret AD as a binary classification problem, we rely on the typical assumption that most of the unlabeled training data is normal by assigning $y = +1$ to all unlabeled examples. Already labeled normal examples and labeled anomalies retain their assigned labels of $\tilde{y} = +1$ and $\tilde{y} = -1$ respectively. We train the supervised classifier on the binary cross-entropy loss. Note that in scenario (i), in particular, the supervised classifier has perfect, unpolluted label information but still fails to generalize as there are novel anomaly classes at testing.

SGD Optimization Details for Deep Methods We use the Adam optimizer with recommended default hyperparameters [32] and apply Batch Normalization [29] in SGD optimization. For all deep approaches and on all datasets, we employ a two-phase ("searching" and "fine-tuning") learning rate schedule. In the searching phase we first train with a learning rate $\varepsilon = 10^{-4}$ for 50 epochs. In the fine-tuning phase we train with $\varepsilon = 10^{-5}$ for another 100 epochs. We always use a batch size of 200. For the autoencoder, SS-DGM, and the supervised classifier, we initialize the network with uniform Glorot weights [21]. For Deep SVDD and Deep SAD, we establish an unsupervised pre-training routine via autoencoder as explained in Appendix A. We set the network ϕ to be the encoder of the autoencoder that we train beforehand.

D Complete Experimental Results

Besides Tables 3–6 that list the complete experimental results of all the methods, we provide AUC scatterplots of the best (1^{st}) vs. second best (2^{nd}) performing methods in the experimental scenarios (i)–(iii) on the most complex CIFAR-10 dataset. If many points fall above the identity line, this is a strong indication that the best method indeed significantly outperforms the second best, which is often the case for Deep SAD.



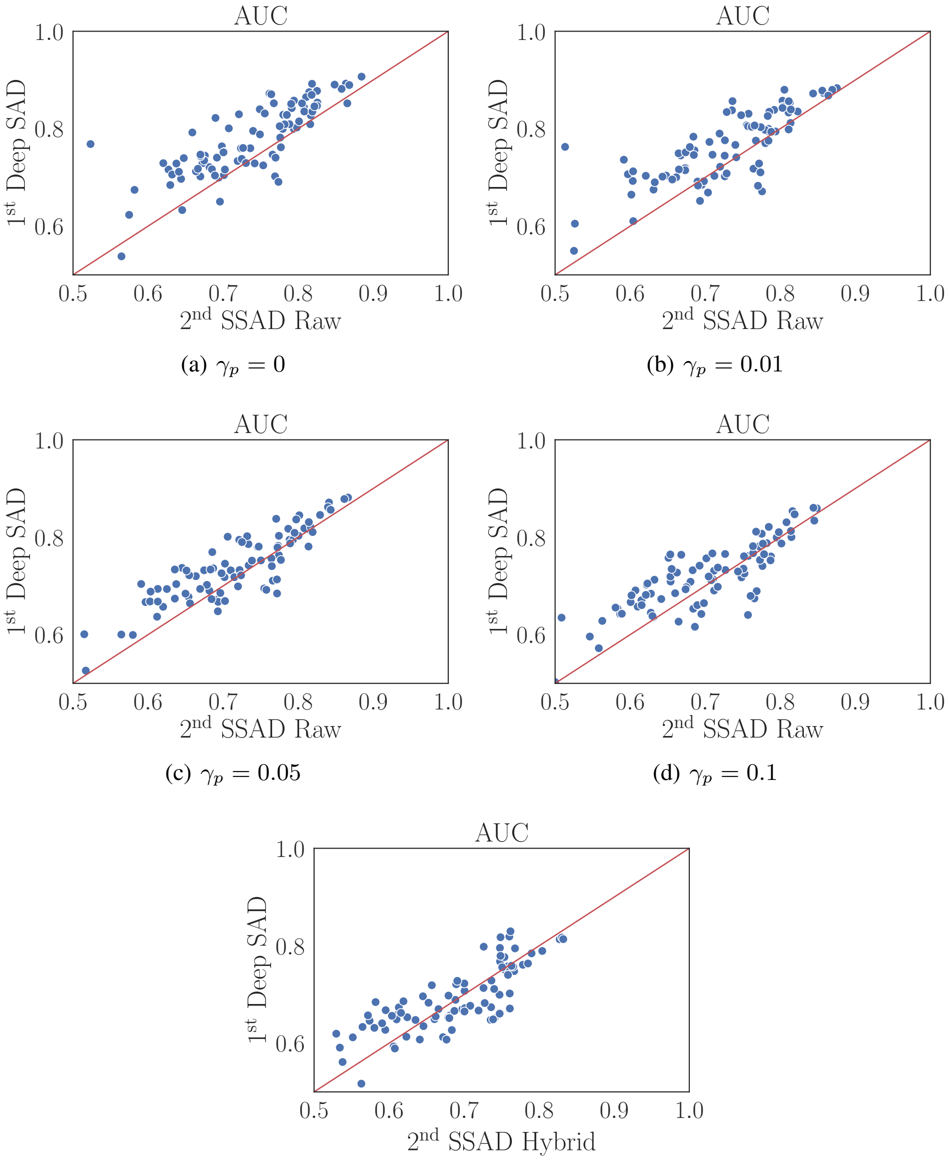
(a) $\gamma_l = 0.01$

(b) $\gamma_l = 0.05$

(c) $\gamma_l = 0.1$

(d) $\gamma_l = 0.2$

Figure 6: AUC scatterplot of best (1^{st}) vs. second best (2^{nd}) performing method in experimental scenario (i) on CIFAR-10, where we increase the ratio of labeled anomalies γ_l in the training set.



(a) $\gamma_p = 0$

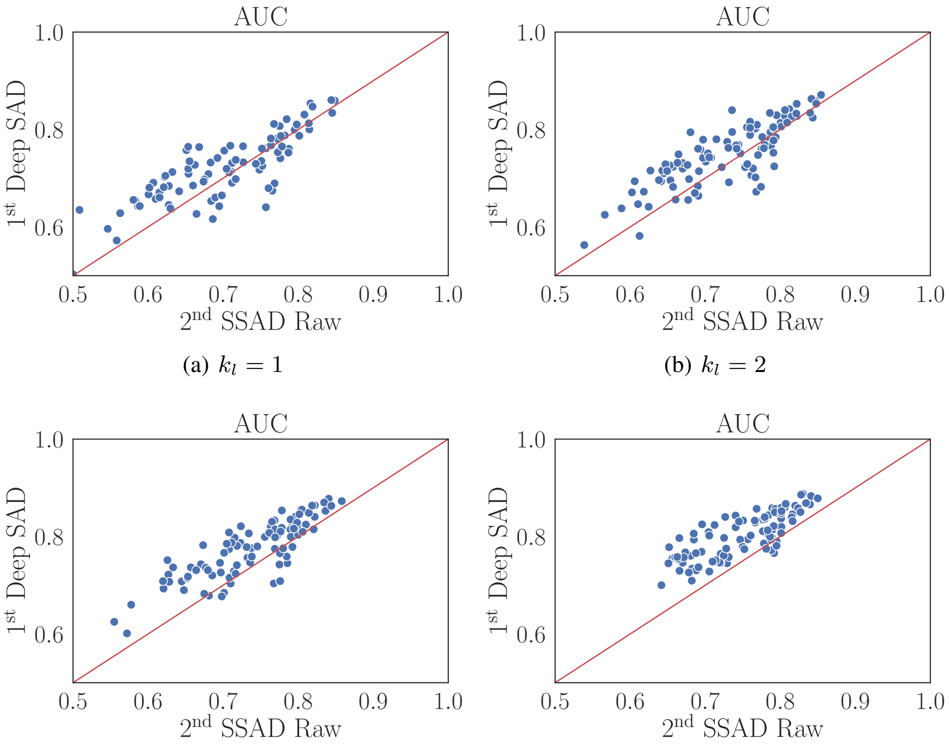
(b) $\gamma_p = 0.01$

(c) $\gamma_p = 0.05$

(d) $\gamma_p = 0.1$

(e) $\gamma_p = 0.2$

Figure 7: AUC scatterplot of best (1^{st}) vs. second best (2^{nd}) performing method in experimental scenario (ii) on CIFAR-10, where we pollute the unlabeled part of the training set with (unknown) anomalies with various ratios γ_p.



(a) $k_l = 1$

(b) $k_l = 2$

(c) $k_l = 3$

(d) $k_l = 5$

Figure 8: AUC scatterplot of best (1^{st}) vs. second best (2^{nd}) performing method in experimental scenario (iii) on CIFAR-10, where we increase the number of anomaly classes k_l included in the labeled training data.

Table 3: Complete results of experimental scenario (i), where we increase the ratio of labeled anomalies γ_l in the training set. We report the avg. AUC with st. dev. computed over 90 experiments at various ratios γ_l.

Data	γ_l	OC-SVM Raw	OC-SVM Hybrid	IF Raw	IF Hybrid	KDE Raw	KDE Hybrid	CAE	Deep SVDD	SSAD Raw	SSAD Hybrid	SS-DGM	Deep SAD	Supervised Classifier
MNIST	.00	96.0±2.9	96.3±2.5	85.4±8.7	90.5±5.3	95.0±3.3	87.8±5.6	92.9±5.7	92.8±4.9	96.0±2.9	96.3±2.5		92.8±4.9	
	.01									96.6±2.4	96.8±2.3	89.9±9.2	96.4±2.7	92.8±5.5
	.05									93.3±3.6	97.4±2.0	92.2±5.6	96.7±2.4	94.5±4.6
	.10									90.7±4.4	97.6±1.7	91.6±5.5	96.9±2.3	95.0±4.7
	.20									87.2±5.6	97.8±1.5	91.2±5.6	96.9±2.4	95.6±4.4
F-MNIST	.00	92.8±4.7	91.2±4.7	91.6±5.5	82.5±8.1	92.0±4.9	69.7±14.4	90.2±5.8	89.2±6.2	92.8±4.7	91.2±4.7		89.2±6.2	
	.01									92.1±5.0	89.4±6.0	65.1±16.3	90.0±6.4	74.4±13.6
	.05									88.3±6.2	90.5±5.9	71.4±12.7	90.5±6.5	76.8±13.2
	.10									85.5±7.1	91.0±5.6	72.9±12.2	91.3±6.0	79.0±12.3
	.20									82.0±8.0	89.7±6.6	74.7±13.5	91.0±5.5	81.4±12.0
CIFAR-10	.00	62.0±10.6	63.8±9.0	60.0±10.0	59.9±6.7	59.9±11.7	56.1±10.2	56.2±13.2	60.9±9.4	62.0±10.6	63.8±9.0		60.9±9.4	
	.01									73.0±8.0	70.5±8.3	49.7±1.7	72.6±7.4	55.6±5.0
	.05									71.5±8.1	73.3±8.4	50.8±4.7	77.9±7.2	63.5±8.0
	.10									70.1±8.1	74.0±8.1	52.0±5.5	79.8±7.1	67.7±9.6
	.20									67.4±8.8	74.5±8.0	53.2±6.7	81.9±7.0	80.5±5.9

Table 4: Complete results of experimental scenario (ii), where we pollute the unlabeled part of the training set with (unknown) anomalies. We report the avg. AUC with st. dev. computed over 90 experiments at various ratios γ_p.

		OC-SVM	**OC-SVM**	**IF**	**IF**	**KDE**	**KDE**		**Deep**	**SSAD**	**SSAD**		**Deep**	**Supervised**
Data	γ_p	**Raw**	**Hybrid**	**Raw**	**Hybrid**	**Raw**	**Hybrid**	**CAE**	**SVDD**	**Raw**	**Hybrid**	**SS-DGM**	**SAD**	**Classifier**
MNIST	.00	96.0±2.9	96.3±2.5	85.4±8.7	90.5±5.3	95.0±3.3	87.8±5.6	92.9±5.7	92.8±4.9	97.9±1.8	97.4±2.0	92.2±5.6	96.7±2.4	94.5±4.6
	.01	94.3±3.9	95.6±2.5	85.2±8.8	90.6±5.0	91.2±4.9	87.9±5.3	91.3±6.1	92.1±5.1	96.6±2.4	95.2±2.3	92.0±6.0	95.5±3.3	91.5±5.9
	.05	91.4±5.2	93.8±3.9	83.9±9.2	89.7±6.0	85.5±7.1	87.3±7.0	87.2±7.1	89.4±5.8	93.4±3.4	89.5±3.9	91.0±6.9	93.5±4.1	86.7±7.4
	.10	88.8±6.0	91.4±5.1	82.3±9.5	88.2±6.5	82.1±8.5	85.9±6.6	83.7±8.4	86.5±6.8	90.7±4.4	86.0±4.6	89.7±7.5	91.2±4.9	83.6±8.2
	.20	84.1±7.6	85.9±7.6	78.7±10.5	85.3±7.9	77.4±10.9	82.6±8.6	78.6±10.3	81.5±8.4	87.4±5.6	82.1±5.4	87.4±8.6	86.6±6.6	79.7±9.4
F-MNIST	.00	92.8±4.7	91.2±4.7	91.6±5.5	82.5±8.1	92.0±4.9	69.7±14.4	90.2±5.8	89.2±6.2	94.0±4.4	90.5±5.9	71.4±12.7	90.5±6.5	76.8±13.2
	.01	91.7±5.0	91.5±4.6	91.5±5.5	84.9±7.2	89.4±6.3	73.9±12.4	87.1±7.3	86.3±6.3	92.2±4.9	87.8±6.1	71.2±14.3	87.2±7.1	67.3±8.1
	.05	90.7±5.5	90.7±4.9	90.9±5.9	85.5±7.2	85.2±9.1	75.4±12.9	81.6±9.6	80.6±7.1	88.3±6.2	82.7±7.8	71.9±14.3	81.5±8.5	59.8±4.6
	.10	89.5±6.1	89.3±6.2	90.2±6.3	85.5±7.7	81.8±11.2	77.8±12.0	77.4±11.1	76.2±7.3	85.6±7.0	79.8±9.0	72.5±15.5	78.2±9.1	56.7±4.1
	.20	86.3±7.7	88.1±6.9	88.4±7.6	86.3±7.4	77.4±13.6	82.1±9.8	72.5±12.6	69.3±6.3	81.9±8.1	74.3±10.6	70.8±16.0	74.8±9.4	53.9±2.9
CIFAR-10	.00	62.0±10.6	63.8±9.0	60.0±10.0	59.9±6.7	59.9±11.7	56.1±10.2	56.2±13.2	60.9±9.4	73.8±7.6	73.3±8.4	50.8±4.7	77.9±7.2	63.5±8.0
	.01	61.9±10.6	63.8±9.3	59.9±10.1	59.9±6.7	59.2±12.3	56.3±10.4	56.2±13.1	60.5±9.4	73.0±8.0	72.8±8.1	51.1±4.7	76.5±7.2	62.9±7.3
	.05	61.4±10.7	62.6±9.2	59.6±10.1	59.6±6.4	58.1±12.9	55.6±10.5	55.7±13.3	59.6±9.8	71.5±8.2	71.0±8.4	50.1±2.9	74.0±6.9	62.2±8.2
	.10	60.8±10.7	62.9±8.2	58.8±10.1	59.1±6.6	57.3±13.5	54.9±11.1	55.4±13.3	58.6±10.0	69.8±8.4	69.3±8.5	50.5±3.6	71.8±7.0	60.6±8.3
	.20	60.3±10.3	61.9±8.1	57.9±10.1	58.3±6.2	56.2±13.9	54.2±11.1	54.6±13.3	57.0±10.6	67.8±8.6	67.9±8.1	50.1±1.7	68.5±7.1	58.5±6.7

Table 5: Complete results of experimental scenario (iii), where we increase the number of anomaly classes k_l included in the labeled training data. We report the avg. AUC with st. dev. computed over 100 experiments at various numbers k_l.

Data	k_l	**OC-SVM** **Raw**	**OC-SVM** **Hybrid**	**IF** **Raw**	**IF** **Hybrid**	**KDE** **Raw**	**KDE** **Hybrid**	**CAE**	**Deep** **SVDD**	**SSAD** **Raw**	**SSAD** **Hybrid**	**SS-DGM**	**Deep** **SAD**	**Supervised** **Classifier**
MNIST	0	88.8±6.0	91.4±5.1	82.3±9.5	88.2±6.5	82.1±8.5	85.9±6.6	83.7±8.4	86.5±6.8	88.8±6.0	91.4±5.1		86.5±6.8	
	1									90.7±4.4	86.0±4.6	89.7±7.5	91.2±4.9	83.6±8.2
	2									92.5±3.6	87.7±3.8	92.8±5.3	92.0±3.6	90.3±4.6
	3									93.9±3.3	89.8±3.3	94.9±4.2	94.7±2.8	93.9±2.8
	5									95.5±2.5	91.9±3.0	96.7±2.3	97.3±1.8	96.9±1.7
F-MNIST	0	89.5±6.1	89.3±6.2	90.2±6.3	85.5±7.7	81.8±11.2	77.8±12.0	77.4±11.1	76.2±7.3	89.5±6.1	89.3±6.2		76.2±7.3	
	1									85.6±7.0	79.8±9.0	72.5±15.5	78.2±9.1	56.7±4.1
	2									87.8±6.1	80.1±10.5	74.3±15.4	80.5±8.2	62.3±2.9
	3									89.4±5.5	83.8±9.4	77.5±14.7	83.9±7.4	67.3±3.0
	5									91.2±4.8	86.8±7.7	79.9±13.8	87.3±6.4	75.3±2.7
CIFAR-10	0	60.8±10.7	62.9±8.2	58.8±10.1	59.1±6.6	57.3±13.5	54.9±11.1	55.4±13.3	58.6±10.0	60.8±10.7	62.9±8.2		58.6±10.0	
	1									69.8±8.4	69.3±8.5	50.5±3.6	71.8±7.0	60.6±8.3
	2									73.0±7.1	72.3±7.5	50.3±2.4	75.2±6.4	61.0±6.6
	3									73.8±6.6	73.3±7.0	50.0±0.7	77.5±5.9	62.7±6.8
	5									75.1±5.5	74.2±6.5	50.0±1.0	80.4±4.6	60.9±4.6

Table 6: Complete results on classic AD benchmark datasets in the setting with no pollution $\gamma_p = 0$ and a ratio of labeled anomalies of $\gamma_l = 0.01$ in the training set. We report the avg. AUC with st. dev. computed over 10 seeds.

Data	OC-SVM Raw	OC-SVM Hybrid	CAE	Deep SVDD	SSAD Raw	SSAD Hybrid	SS-DGM	Deep SAD	Supervised Classifier
arrhythmia	84.5±3.9	76.7±6.2	74.0±7.5	74.6±9.0	86.7±4.0	78.3±5.1	50.3±9.8	75.9±8.7	39.2±9.5
cardio	98.5±0.3	82.8±9.3	94.3±2.0	84.8±3.6	98.8±0.3	86.3±5.8	66.2±14.3	95.0±1.6	83.2±9.6
satellite	95.1±0.2	68.6±4.8	80.0±1.7	79.8±4.1	96.2±0.3	86.9±2.8	57.4±6.4	91.5±1.1	87.2±2.1
satimage-2	99.4±0.8	96.7±2.1	99.9±0.0	98.3±1.4	99.9±0.1	96.8±2.1	99.2±0.6	99.9±0.1	99.9±0.1
shuttle	99.4±0.9	94.1±9.5	98.2±1.2	86.3±7.5	99.6±0.5	97.7±1.0	97.9±0.3	98.4±0.9	95.1±8.0
thyroid	98.3±0.9	91.2±4.0	75.2±10.2	72.0±9.7	97.9±1.9	95.3±3.1	72.7±12.0	98.6±0.9	97.8±2.6