SELF-SUPERVISE, REFINE, REPEAT: IMPROVING UNSUPERVISED ANOMALY DETECTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Anomaly detection (AD) – separating anomalies from normal data – has many applications across domains, from manufacturing to healthcare. While most previous works have been shown to be effective for cases with fully or partially labeled data, that setting is in practice less common due to labeling being particularly tedious for this task. In this paper, we focus on fully *unsupervised* AD, in which the entire training dataset, containing both normal and anomalous samples, is unlabeled. To tackle this problem effectively, we propose to improve the robustness of one-class classification trained on self-supervised representations using a data refinement process. Our proposed data refinement approach is based on an ensemble of one-class classifiers (OCCs), each of which is trained on a disjoint subset of training data. Representations learned by self-supervised learning on the refined data are iteratively updated as the refinement improves. We demonstrate our method on various unsupervised AD tasks with image and tabular data. With a 10% anomaly ratio on CIFAR-10 image data / 2.5% anomaly ratio on Thyroid tabular data, the proposed method outperforms the state-of-the-art one-class classification method by 6.3 AUC and 12.5 average precision / 22.9 F1-score.

1 INTRODUCTION

Anomaly detection (AD), the task of distinguishing anomalies from normal data, plays a crucial role in many real-world applications such as detecting faulty products using visual sensors in manufacturing, fraudulent behaviors at credit card transactions, or adversarial outcomes at intensive care units.

AD has been considered under various settings based on the availability of negative (normal) and positive (anomalous) data and their labels at training, as overviewed in Sec. 2. Each application scenario is dominated by different challenges. When entire positive and negative data are available along with their labels (Fig. 1a), the problem can be treated as supervised classification and the dominant challenge becomes the imbalance in label distributions [4, 15, 19, 26, 31, 34]. When only negative labeled data are available (Fig. 1b), the problem is 'one-class classification' [22, 25, 32, 43, 45, 49, 51]. Various works have also extended approaches designed for these to settings with additional unlabeled data (Fig. 1c,d,e) [10, 17, 23, 44, 55] in a semi-supervised setting. While there exist many prior works in these settings, *they all depend on some labeled data*, which is not desirable in all application scenarios.

Unsupervised AD, on the other hand, poses unique challenges in the absence of any labeled data information, and a straightforward adaption of methods developed with the assumption of labeled data would be suboptimal. For example, some recent studies [7, 57] have applied one-class classifiers (OCCs) that are known to yield impressive performance when trained on negative samples [7, 22, 25, 32, 49] to unsupervised AD, but their performance for unsupervised AD has been quite sub-optimal. Fig. 2 illustrates this, showing the unsupervised AD performance of state-of-the-art Deep OCCs [49] with different anomaly ratios in unlabeled training data – the average precision significantly drops even when a small portion (2%) of training data is contaminated with anomalies.

Our framework SRR (**S**elf-supervise, **R**efine, **R**epeat), overviewed in Fig. 3, brings a novel approach to unsupervised AD with the principles of self-supervised learning without labels and iterative data refinement based on the agreement of OCC outputs. We propose to improve the state-of-the-art performance of OCCs, e.g. [32, 49], by refining the unlabeled training data so as to address the fundamental challenges elaborated above. SRR iteratively trains deep representations using refined

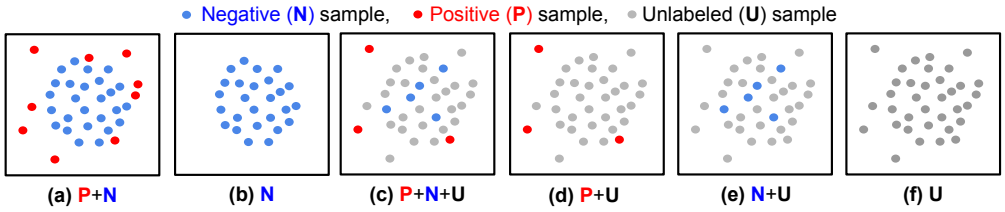


Figure 1: AD problem settings. Blue and red dots are for **labeled** negative (normal) and positive (anomalous) samples, respectively. Grey dots denote **unlabeled** samples. While previous works mostly focus on supervised (a, b) or semi-supervised (c, d, e) settings, we tackle an AD problem using only unlabeled data (f) that may contain both negative and positive samples.

data while improving the refinement of unlabeled data by excluding potentially-positive (anomalous) samples. For the data refinement process, we employ an ensemble of OCCs, each of which is trained on a disjoint subset of unlabeled training data. The samples are declared as normal if there is a consensus between all the OCCs. The refined training data are used to train the final OCC to generate the anomaly scores in the unsupervised setting. Most prior unsupervised AD works [22, 25, 32, 49] assume that the data contains entirely negative samples, which makes them not truly unsupervised as they require having humans to do the data filtering. Similar to ours, some prior unsupervised AD works [7, 43, 57] considered evaluating on an unsupervised setting where there exist a small percentage of anomalous samples in the training data, i.e. operating in 'truly' unsupervised setting without having the need for humans to do any filtering in the training data. However, these methods often suffered from significant performance degradation as the ratio of anomalous sample ratio has increased (see Sec. 4.2). Our framework SRR aims to provide robustness in performance as the anomalous sample ratio increases, as shown in Fig. 2.

We conduct extensive experiments across various datasets from different domains, including semantic AD (CIFAR-10 [28], Dog-vs-Cat [18]), real-world manufacturing visual AD use case (MVTec [8]), and tabular AD benchmarks. We consider methods with both shallow [35, 45] and deep [7, 32, 49] models. We evaluate models at different anomaly ratios of unlabeled training data and show that SRR significantly boosts performance. For example, in Fig. 2, SRR improves more than 15.0 average precision (AP) with a 10% anomaly ratio compared to a state-of-the-art one-class contrastive representation model [49] on CIFAR-10. Similarly, on MVTec SRR retains a strong performance, dropping less than 1.0 AUC with 10% anomaly ratio, while the best existing OCC [32] drops more than 6.0 AUC. We further investigate the efficacy of our design choices, such as the number of ensemble classifiers, thresholds, and refinement of deep representations via ablation studies.

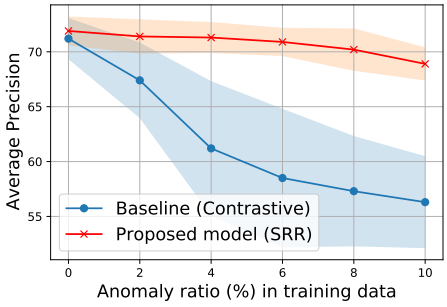


Figure 2: Performance of our proposed model and a baseline OCC using contrastive representation [49] on CIFAR-10 with different anomaly ratios in the training data.

2 Related Work

There are various existing works under the different settings described in Fig. 1:

The positive + negative setting is often considered as a supervised binary classification problem. The challenge arises due to the imbalance in label distributions as positive (anomalous) samples are rare. As summarized in [12], to address this, over-/under-sampling [15, 19], weighted optimization [4, 26], synthesizing data of minority classes [31, 34], and hybrid methods [2, 21] have been studied.

The negative setting is often converted to a one-class classification problem, with the goal of finding a decision boundary that includes as many one-class samples as possible. Shallow models for this setting include one-class support vector machines [45] (OC-SVM), support vector data description [51] (SVDD), kernel density estimation (KDE) [30], and Gaussian density estimation (GDE) [42]. There are also auto-encoder based models [56] that treat the reconstruction error as the

anomaly score. Deep learning based OCCs have been developed, such as Deep OCC [43], geometric transformation [22], or outlier exposure [25]. Noting the degeneracy or inconsistency of learning objectives of existing end-to-end trainable Deep OCCs, [49] proposed a deep representation OCC, a two-stage framework that learns self-supervised representations [16, 27] followed by shallow OCCs. That work was extended for texture anomaly localization with CutPaste [32]. The robustness to very low anomaly ratios of these methods under the unsupervised setting was explored in [7, 57].

The semi-supervised setting is defined as utilizing a small set of labeled samples and large set of unlabeled samples to distinguish anomalies from normal data. Depending on which labeled samples are given, this setting can be split into three sub-categories. When only some positive/negative labeled samples are provided, we denote that as a PU/NU setting. Most previous works in semi-supervised AD settings focus on the NU setting where only some of the normal labeled samples are given [1, 40, 50]. The PNU setting is a more general semi-supervised setting where subsets of both positive and negative labeled samples are given. Deep SAD [44] and SU-IDS [38] are included in this category. We show the significant outperformance of our proposed SRR framework compared to Deep SAD, for multiple benchmark datasets when not using any labeled data (see Sec. 4.2).

The unlabeled setting has received relatively less attention despite its significance in automating machine learning. The popular methods for this setting include isolation forest [35] and local outlier factor [13]. However, they are difficult to scale, and less compatible with recent advances in representation learning. While OCCs, such as OC-SVM, SVDD, or their deep counterparts, apply to unlabeled settings by assuming the data is all negative, and the robustness of those methods has also been demonstrated in part [7, 57], in practice we observe a significant performance drop with a high anomaly ratio, shown in Fig. 2. In contrast, our proposed framework is able to maintain high performance across anomaly ratios.

Data refinement has been applied to AD in some prior works. [5, 41] generate pseudo-labels using binary classification and OC-SVM for data refinement to boost the consequent AD performances in unsupervised settings. [6, 29, 39, 52, 56] used the reconstruction errors of the auto-encoder as an indicator for removing possible anomalies. [20] and [37] used data refinement for AD in supervised and semi-supervised settings.

Self-training [36, 46] is an iterative training mechanism using predicted pseudo labels as targets for model training. It has regained popularity recently with its successful results in semi-supervised image classification [9, 48, 54]. To improve the quality of pseudo labels, employment of an ensemble of classifiers has also been studied. [14] trains an ensemble of classifiers with different classification methods to make a consensus for noisy label verification. Co-training [11] trains multiple classifiers, each of which is trained on the distinct views, to supervise other classifiers. Co-teaching [24] and DivideMix [33] share a similar idea in that they both train multiple deep networks on separate data batches to learn different decision boundaries, thus becoming useful for noisy label verification. While sharing a similarity, the proposed framework has clear differences from the previous works. SRR performs *iterative training* with data refinement (with robust *ensemble* methods) and self-supervised learning for *unsupervised* learning of an anomaly detector.

3 Proposed Framework

Self-supervise, **R**efine, and **R**epeat (SRR) is an iterative training framework, where we refine the data (Sec. 3.1) and update the representation with the refined data (Sec. 3.2), followed by OCCs on refined representations. Fig. 3 overviews the framework and Algorithm 1 provides the pseudo code.

Notation. We denote the training data as $\mathcal{D} = \{\mathbf{x}_i\}_{i=1}^{N}$ where $\mathbf{x}_i \in \mathcal{X}$ and N is the number of training samples. $y_i \in \{0, 1\}$ is the corresponding label to $\mathbf{x}_i$, where 0 denotes normal (negative) and 1 denotes anomaly (positive). Note that labels are not provided in the unsupervised setting.

Let us denote a feature extractor as $g : \mathcal{X} \to \mathcal{Z}$. g may include any data preprocessing functions, an identity function (if raw data is directly used for one-class classification), and learned or learnable representation extractors such as deep neural networks. Let us define an OCC as $f : \mathcal{Z} \to [-\infty, \infty]$ that outputs anomaly scores given the input features $g(\mathbf{x})$. The higher the score $f(g(\mathbf{x}))$, the more anomalous the sample $\mathbf{x}$ is. The binary anomaly prediction is made after thresholding: $\mathbb{1}\big(f(g(\mathbf{x})) \geq \eta\big)$.

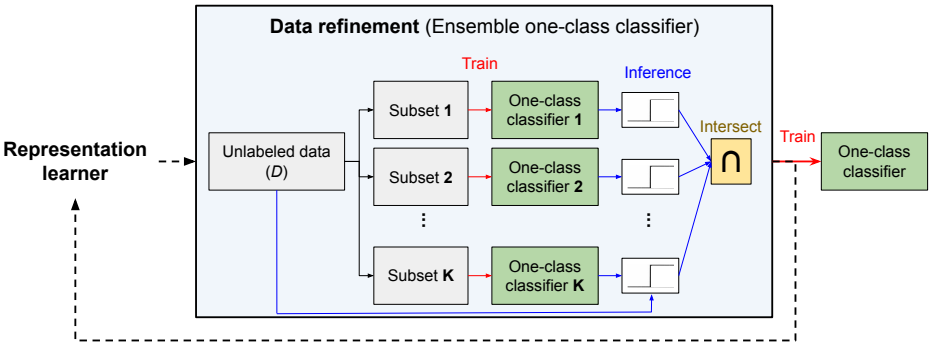


Figure 3: Block diagram of SRR composed of representation learner (Sec. 3.2), data refinement (Sec. 3.1), and final OCC blocks. The representation learner updates the deep models using refined data from the data refinement block. Data refinement is done by an ensemble of OCCs, each of which is trained on K disjoint subsets of unlabeled training data. Samples predicted as normal by all classifiers are retained in the refined data, and are used to update the representation learner and final OCC. The process is repeated iteratively until convergence.

3.1 Data Refinement

A naive way to generate pseudo labels of unlabeled data is to construct an OCC on raw data or learned representations as in [49] and threshold the anomaly score to obtain a binary label for normal vs. anomalous. As we update the model with refined data that excludes samples that are predicted to be anomalous, it is important to generate pseudo labels of training data as accurately as possible.

To this end, instead of training a single classifier (unlike most previous works on data refinement for AD [5, 41, 56]), we train an ensemble of K OCCs and aggregate their predictions to generate pseudo labels. We illustrate the data refinement block in Fig. 3 and as RefineData in Algorithm 1. Specifically, we randomly divide the unlabeled training data $\mathcal{D}$ into K disjoint subsets $\mathcal{D}_1, ..., \mathcal{D}_K$, and train K different OCCs $(f_1, ..., f_K)$ on corresponding subsets $(\mathcal{D}_1, ..., \mathcal{D}_K)$. Then, we estimate a binary pseudo-label of the data $\mathbf{x}_i \in \mathcal{D}$ as follows:

$$\hat{y}_i = 1 - \prod_{k=1}^{K} \Big[1 - \mathbb{1}\big(f_k(g(\mathbf{x}_i)) \geq \eta_k\big)\Big] \tag{1}$$

$$\eta_k = \max \eta \ \text{ s.t. } \ \frac{1}{N} \sum_{i=1}^{N} \mathbb{1}\big(f_k(g(\mathbf{x}_i)) \geq \eta\big) \geq \gamma \tag{2}$$

where $\mathbb{1}(\cdot)$ is the indicator function that outputs 1/0 if the input is True/False. $f_k(g(\mathbf{x}_i))$ represents an anomaly score of $\mathbf{x}_i$ for an OCC f_k. η_k in equation 2 is a threshold determined as a γ percentile of the anomaly score distribution $\{f_k(g(\mathbf{x}_i))\}_{i=1}^{N}$.

To interpret equation 1, $\mathbf{x}_i$ is predicted as normal, i.e. $\hat{y}_i = 0$, if all K OCCs predict it as normal. While this may be too strict and potentially reject many true normal samples in the training set, we find that empirically, it is critical to be able to exclude true anomalous samples from the training set. The effectiveness of the employment of an ensemble of classifiers is empirically shown in Sec. 4.3.

3.2 Representation Update

SRR follows the idea of deep representation OCCs [49], where in the first stage a deep neural network is trained with self-supervised learning (such as rotation prediction [22], contrastive [49], or CutPaste [32]) to obtain meaningful representations of the data, and in the second stage OCCs are trained on these learned representations. Such a two-stage framework is shown to be beneficial as it prevents the 'hypersphere collapse' of the deep OCCs by the favorable inductive bias it brings with the architectural constraints [43].

Here, we propose to conduct self-supervised representation learning jointly with data refinement. More precisely, we train a feature extractor g using $\hat{\mathcal{D}} = \{\mathbf{x}_i \mid \hat{y}_i = 0\}$, a subset of unlabeled data $\mathcal{D}$ that only includes samples whose predicted labels with an ensemble OCC from Sec. 3.1 are

Algorithm 1 SRR: Self-supervise, Refine, Repeat.

Input: Training data $\mathcal{D} = \{\mathbf{x}_i\}_{i=1}^N$, Ensemble count ($K$), threshold ($\gamma$)
Output: Refined data ($\hat{\mathcal{D}}$), trained OCC (f), feature extractor (g)

1: **function** REFINEDATA($\mathcal{D}, g, K, \gamma$)
2: Train OCC models $\{f_k\}_{k=1}^K$ on $\{\mathcal{D}_k\}_{k=1}^K$, K disjoint subsets of the training data $\mathcal{D}$.
3: Compute thresholds η_k's for γ percentile of anomaly distributions (equation 2).
4: Predict binary labels $\hat{y}_i$ (equation 1).
5: Return $\hat{\mathcal{D}} = \{\mathbf{x}_i : \hat{y}_i = 0, \mathbf{x}_i \in \mathcal{D}\}$.
6: **end function**
7: **function** SRR($\mathcal{D}, K, \gamma$)
8: Initialize the feature extractor g.
9: **while** g not converged **do**
10: $\hat{\mathcal{D}}$ = REFINEDATA($\mathcal{D}, g, K, \gamma$).
11: Update g using $\hat{\mathcal{D}}$ with self-supervised learning objectives.
12: **end while**
13: $\hat{\mathcal{D}}$ = REFINEDATA($\mathcal{D}, g, K, \gamma$).
14: Train an OCC model (f) on refined data ($\hat{\mathcal{D}}$).
15: **end function**

negative. We also update $\hat{\mathcal{D}}$ as we proceed with representation learning. The proposed method is illustrated in Algorithm 1 as SRR. In contrast to previous works [32, 49] that use the entire training data for learning self-supervised representation, we find it necessary to refine the training data even for learning deep representations. Without representation refinement, the performance improvements of SRR are limited, as shown in Sec. 4.3.2. Last, for test-time prediction, we train an OCC on refined data $\hat{\mathcal{D}}$ using updated representations by g as in line 13-14 in Algorithm 1.

3.3 UNSUPERVISED MODEL SELECTION

As SRR is designed for unsupervised AD, labeled validation data for hyperparameter tuning is typically not available and *the framework should enable robust model selection without any reliance on labeled data*. Here, we provide insights on how to select important hyperparameters, and later in Sec. 4.3.1 perform sensitivity analyses for these hyperparameters.

Data refinement of SRR introduces two hyperparameters: the number of OCCs (K) and the percentile threshold (γ). There is a trade-off between the number of classifiers for the ensemble and the size of disjoint subsets for training each classifier. With large K, we aggregate prediction from many classifiers, each of which may contain randomness from training. This comes at a cost of reduced performance per classifier as we use smaller subsets to train them. In practice, we find $K = 5$ works well across different datasets and anomaly ratios. γ controls the purity and coverage of refined data. If γ is large, and thus classifiers reject too many samples, the refined data could be more pure and contain mostly the normal samples; however, the coverage of the normal samples would be limited. On the other hand, with a small γ, the refined data may still contain many anomalies and the performance improvement with SRR would be limited. We empirically observe that SRR is robust to the selection of γ when it is chosen from a reasonable range. In our empirical experiments, we find $\sim 1 - 2\times$ of the true anomaly ratio to be a reasonable choice. In other words, it is safer to use γ higher than the expected true anomaly ratio. In some cases, the true anomaly ratio may not be available at all; for such scenarios, we propose Otsu's method [47] to estimate the anomaly ratio of the training data for determining the threshold γ (experimental results are in the Appendix A.5).

4 EXPERIMENTS

We evaluate the efficacy of our proposed framework for unsupervised AD tasks on tabular (Sec. 4.1) and image (Sec. 4.2) data types. We experiment varying ratios of anomaly samples in unlabeled training data and with different combinations of representation learning and OCCs. In Sec. 4.3, we provide performance analyses to better explain major constituents of the performance, as well as sensitivity to hyperparameter values.

Implementation details: To reduce the computational complexity of the data refinement block, we utilize a simple OCC such as GDE in the data refinement block. In a two-stage model, we only update the data refinement block at 1st, 2nd, 5th, 10th, 20th, 50th, 100th, 500th epochs instead of every epoch. After 500 epochs, we update the data refinement block per each 500th epoch. Each run of experiments requires a single V100 GPU.

4.1 Experiments on Tabular Data

Datasets. Following [7, 57], we test the efficacy of SRR on a variety of tabular datasets, including KDDCup, Thyroid, or Arrhythmia from the UCI repository [3]. We also use KDDCup-Rev, where the labels of KDDCup are reversed so that an attack represents anomaly [57]. To construct data splits, we use 50% of normal samples for training. In addition, we hold out some anomaly samples (amounting to 10% of the normal samples) from the data. This allows us to simulate unsupervised settings with an anomaly ratio of up to 10% of entire training set. The rest of the data is used for testing.[1] We conduct experiments using 5 random splits and 5 random seeds, and report the average and standard deviation of 25 F1-scores (with scale 0-100) for the performance metric.

Models. We mainly compare with GOAD [7] (the state-of-the-art AD model in the tabular domain) and implement SRR on top of it. GOAD utilizes random transformation classification as the pretext task of self-supervised learning, and the normality score is determined by whether transformations are accurately included in the transformed space of the normal samples. We re-implement GOAD [7] with a few modifications. First, instead of using embeddings to compute the loss, we use a parametric classifier, similarly to augmentation prediction [49]. Second, we follow the two-stage framework [49] to construct deep OCCs. For the N setting, our implementation achieves 98.0 for KDD, 95.0 for KDD-Rev, 75.1 for Thyroid, and 54.8 for Arrhythmia F1-scores, which are comparable to those reported in [7]. Please see Appendix A.3 for formulation and implementation details.

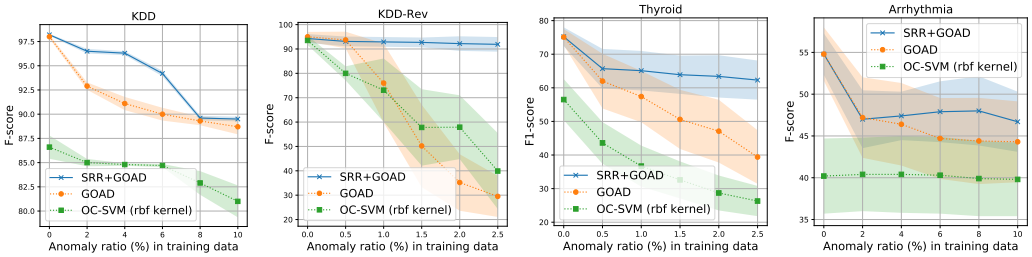


Figure 4: Unsupervised AD performance (F1-score) using OC-SVM (with rbf kernel), GOAD [7], and GOAD with the proposed method SRR on various tabular datasets. Shaded areas represent the standard deviation.

Results. We show results of GOAD (the baseline) and GOAD with SRR in Fig. 4. The ranges of the noise ratio are set to 0% for the anomaly ratios in the original dataset (if anomaly ratios are larger than 10%, we set the maximum anomaly ratio as 10%). For KDD-Rev, we set the maximum anomaly ratio to 2.5% because the performance of GOAD (without SRR) drops significantly even with a small ratio of anomalies in the training data. Fig. 4 shows that integrating SRR significantly improves GOAD (the state-of-the-art methods for tabular OCC). The improvements are more significant especially at higher anomaly ratios. This underlines how SRR can achieve significant improvements for both small (Thyroid & Arrhythmia) and large-scale datasets (KDD & KDD-Rev).

4.2 Experiments on Image Data

Datasets. We evaluate SRR on visual AD benchmarks, including CIFAR-10 [28], f-MNIST [53], Dog-vs-Cat [18], and MVTec [8]. For CIFAR-10, f-MNIST, and Dog-vs-Cat datasets, samples from one class are set to be normal and the rest from other classes are set to be an anomaly. Similar to the experiments on tabular data in Section. 4.1, we swap a certain amount of the normal training data with anomalies given the target anomaly ratio. For MVTec, since there are no anomalous data

[1] Note that the experimental settings with contaminated training data in GOAD [7] and DAGMM [57] are slightly different from ours. Our contamination ratio is defined as the anomaly ratio over the entire training data, while their contamination ratio is the anomaly ratio over all the anomalies in the dataset.

available for training, we borrow 10% of the anomalies from the test set and swap them with normal samples in the training set. Note that 10% of samples borrowed from the test set are excluded from evaluation. For all datasets, we experiment with varying anomaly ratios from 0% to 10%.

We use area under ROC curve (AUC) and average precision (AP) metrics to quantify the performance for visual AD (with scale 0-100). When computing AP, we set the minority class of the test set as label 1 and majority as label 0 (e.g., normal samples are set as label 1 for CIFAR-10 experiments as there are more anomaly samples that are from 9 classes in the test set). We run all experiments with 5 random seeds and report the average performance for each dataset across all classes. Per-class AUC and AP are reported in Appendix A.4.

Models. For semantic AD benchmarks, CIFAR-10, f-MNIST, and Dog-vs-Cat, we compare the SRR with two-stage OCCs [49] using various representation learning methods, such as distribution-augmented contrastive learning [49], rotation prediction [22, 27] and its improved version [49], and denoising autoencoder. For MVTec benchmarks, we use CutPaste [32] as a baseline and compare to its version with SRR integration. For both experiments, we use the ResNet-18 architecture, trained from random initialization, using the hyperparameters from [49] and [32]. The same model and hyperparameter configurations are used for SRR with $K = 5$ classifiers in the ensemble. We set γ as twice the anomaly ratio of training data. For 0% anomaly ratio, we set γ as 0.5. Finally, a Gaussian Density Estimator (GDE) on learned representations is used as OCC.

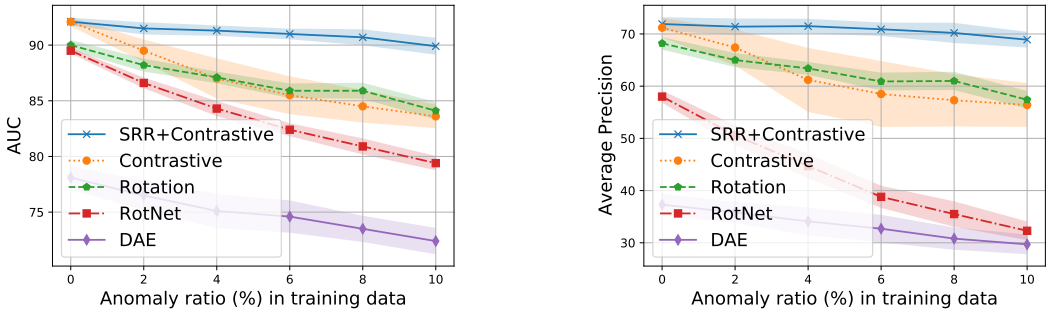


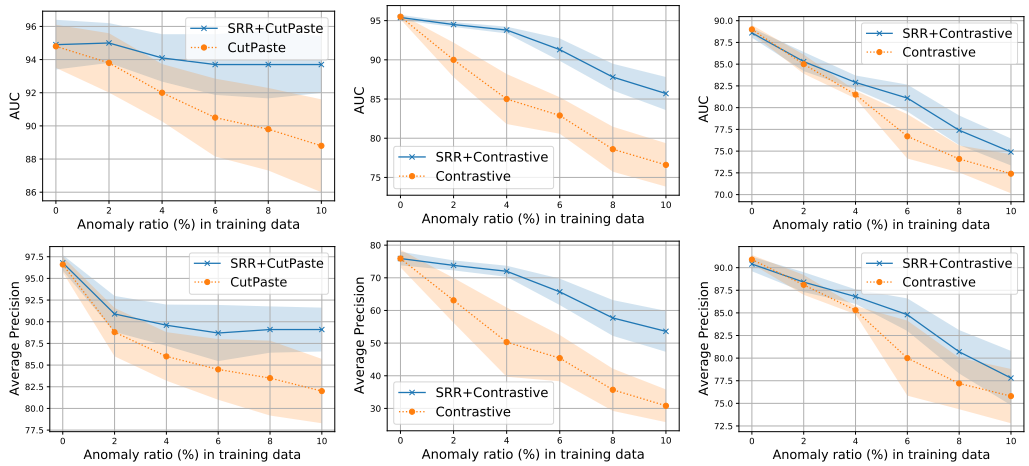
Figure 5: Unsupervised AD performance with various OCCs on CIFAR-10. For SRR we adapt distribution-augmented contrastive representation learning [49]. (Left) AUC, (Right) Average Precision (AP).

Results. Fig. 5 shows a significant performance drop with increased anomaly ratio, regardless of representation learning. For example, the AUC of distribution-augmented contrastive representation [49] drops from 92.1 to 83.6 when anomaly ratio becomes 10%. Similarly, the improved rotation prediction representation [27] drops from 90.0 to 84.1. On the other hand, SRR effectively handles the contamination in the training data and achieves 89.9 AUC with 10% anomaly ratio, reducing the performance drop by 74.1%. An 'oracle' upper bound would be the removal of all anomalies from the training data (which is the same as the performance at 0% anomaly ratio for the same size of the data). As Fig. 5 shows, the performance of SRR is similar to this oracle upper bound performance (less than 2.5 AUC difference) even with high anomaly ratios (10%). The results are also similar in other metrics, such as AP in Fig. 5 or Recall at Precision of 70 and 90 in Fig. 9 in the Appendix.

We repeat experiments on 3 additional visual AD datasets and report results in Fig. 6. We observe consistent and significant improvements over the baseline across different datasets and different one-class classification methods. Note that the improvement is more significant at higher anomaly ratios. For instance, on MVTec dataset, SRR improves AUC by 4.9 and AP by 7.1 compared to the state-of-the-art CutPaste OCC with an anomaly ratio of 10%. In the Appendix (Sec. A.4), we also illustrate per-class performance of SRR compared with the state-of-the-art.

4.3 Performance Analyses

In this section, we conduct sensitivity analyses on two hyperparameters of SRR, namely, the number K for ensemble OCCs and the percentile threshold γ that determines normal and anomalies in the data refinement module. In addition, we show the importance of updating the representation with



(a) MVTec (b) f-MNIST (c) Dog-vs-Cat

Figure 6: Unsupervised AD performance on (a) MVTec (b) f-MNIST, and (c) Dog-vs-Cat datasets with varying anomaly ratios. We use state-of-the-art one-class classification models for baselines, such as distribution-augmented contrastive representations [49] for f-MNIST and Dog-vs-Cat, or CutPaste [32] for MVTec, and build SRR on top of them.

refined data. Ablation studies are conducted on two visual AD benchmarks, CIFAR-10 and MVTec. More results can be found in the Appendix.

4.3.1 Sensitivity to Hyperparameters

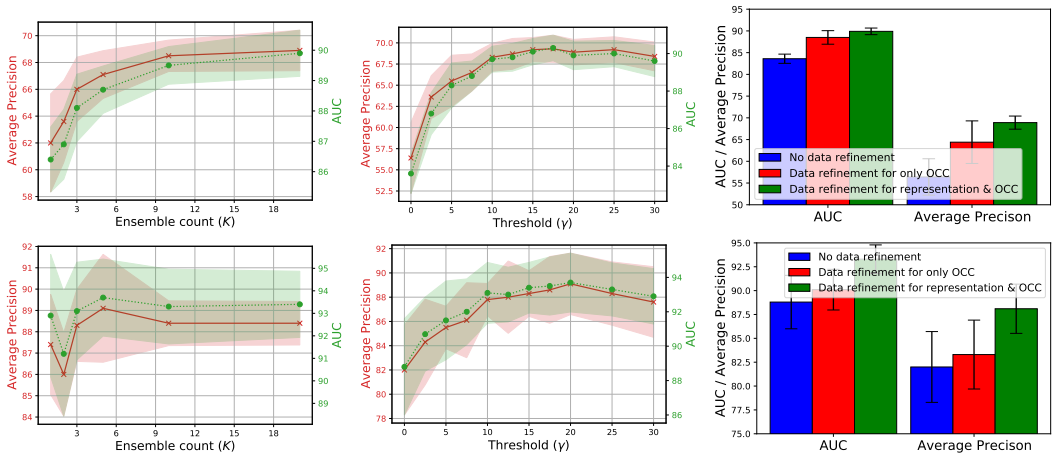
SRR is designed for unsupervised AD and it is important to ensure robust performance against changes in the hyperparameters as model selection without labeled data would be very challenging. Fig. 7 presents the sensitivity analyses of SRR with respect to various hyperparameters. In Fig. 7a, we observe the performance improvement as we increase the number of classifiers for ensemble. This is particularly effective on CIFAR (top in Fig. 7a), where the number of samples in the training data is large enough that even with large K, the number of samples to train each OCC (N/K) would be sufficient. In Fig. 7b, we observe that SRR performs robustly when γ is set to be larger than the actual anomaly ratio (10%). When γ is less than 10%, however, we see a significant drop in performance, all the way to the baseline ($\gamma = 0$). Our results show that SRR improves upon baseline regardless of the threshold. In addition, it suggests that γ could be set to be anywhere from the true anomaly ratio and and 2x the anomaly ratio to maximize its effectiveness. When the true anomaly ratio is unknown, as discussed in Appendix A.5, Otsu's method could be used.

4.3.2 Iteratively Updating Representations with Refined Data

It is possible to decouple representation learning and data refinement of SRR, which would result in a three-stage framework, where we learn representations until convergence without data refinement, followed by the data refinement and learning OCC. Fig. 7c shows that SRR using a pre-trained then fixed representation (i.e. when data refinement is used for OCC only) already improves the performance upon the baseline with no data refinement at any stage of learning representation or classifier. Improvements on CIFAR-10 are 4.9 in AUC and 8.0 in AP; and on MVTec are 1.3 in AD and in 1.3 in AP. This underlines the effectiveness of our data refinement module. More results on applying SRR on raw tabular features or learned (and fixed) visual representations are provided in Appendix A.2. Moreover, we find that learning representation with refined data plays a crucial role, resulting in another improvement in AUC by 1.4, AP by 4.5 on CIFAR-10, and AUC by 3.2, AP by 4.8 on MVTec, upon a SRR using a fixed representation.

4.4 Quantifying the Refinement Efficacy

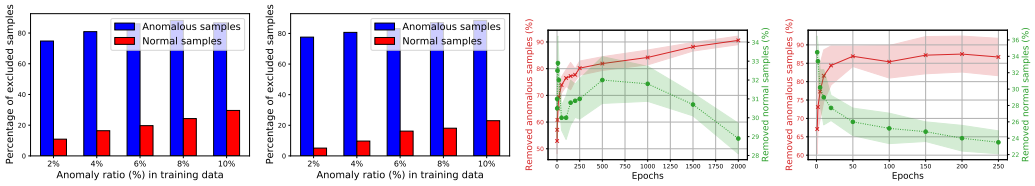
We evaluate how many normal and anomalies are excluded by the proposed data refinement block. As in Fig. 8(a, b), with data refinement, we can exclude more than 80% of anomalies in the training set without removing too many normal samples. For instance, among 4% anomalies in CIFAR-10



(a) Ensemble count K (b) Percentile threshold γ (c) Representation update

Figure 7: Ablation studies on (top) CIFAR-10 and (bottom) MVTec under 10% anomaly ratio setting with respect to (a) ensemble count K, (b) percentile threshold γ, and (c) data refinement with or without representation update.

data, SRR is able to exclude 80% anomalies while removing less than 20% normal samples. Such a high recall of anomalies of SRR is not only useful for unsupervised AD, but also could be useful for improving the annotation efficiency when a budget for active learning is available. Fig. 8(c, d) demonstrates the removed normal and abnormal samples by the data refinement module over training epochs. It shows that better representation learning (as training epochs increase) consistently improves the efficacy of the data refinement.



(a) CIFAR-10 (b) MVTec (c) CIFAR-10 (d) MVTec

Figure 8: Percentage of excluded anomalous and normal samples by data refinement (a, b) with different anomaly ratios in training data and (c, d) over training epochs for 10% anomaly ratio.

5 Conclusion

AD has wide range of practical use cases. A challenging and costly aspect of building an AD system is that anomalies are rare and not easy to detect by humans, making them difficult to label. To this end, we propose a novel AD framework to enable high performance AD without any labels called SRR. SRR can be flexibly integrated with any OCC, and applied on raw data or on a trainable representations. SRR employs an ensemble of multiple OCCs to propose candidate anomaly samples that are refined from training, which allows more robust fitting of the anomaly decision boundaries as well as better learning of data representations. We demonstrate the state-of-the-art AD performance of SRR on multiple tabular and image data.

ETHICS STATEMENT

SRR has the potential to make significant positive impact in real-world AD applications where detecting anomalies is crucial, such as for financial crime elimination, cybersecurity advances, or improving manufacturing quality. We note a potential risk associated with using SRR: when representations are not sufficiently good, there will be a negative cycle of refinement and representation updates/OCC. While we rely on the existence of good representations, in some applications they may be difficult to obtain (e.g., cybersecurity). This paper focuses on the unsupervised setting and demonstrates strong AD performance, opening new horizons for human-in-the-loop AD systems that are low cost and robust. We leave these explorations to future work.

REPRODUCIBILITY STATEMENT

The source code of SRR will be published upon acceptance. For reproducibility, we only use public tabular and image datasets to evaluate the performances of SRR. Complete descriptions of those datasets can be found in **Datasets** subsections in Sec. 4.1 and 4.2. Detailed experimental settings and hyperparameters can be found in **Models** subsections in Sec. 4.1 and 4.2. The implementation details (including hyperparameters that we used) on GOAD [7] are described in Sec. A.3.

REFERENCES

[1] S. Akcay, A. Atapour-Abarghouei, and T. P. Breckon. Ganomaly: Semi-supervised anomaly detection via adversarial training. In *ACCV*, 2018. 3

[2] R. Alejo, R. M. Valdovinos, V. García, and J. H. Pacheco-Sanchez. A hybrid method to face class overlap and class imbalance on neural networks and multi-class scenarios. *PRL*, 34(4):380–388, 2013. 2

[3] A. Asuncion and D. Newman. UCI machine learning repository, 2007. 6

[4] S. Barua, M. M. Islam, X. Yao, and K. Murase. Mwmote–majority weighted minority oversampling technique for imbalanced data set learning. *IEEE Trans on knowledge and data engineering*, 26(2):405–425, 2012. 1, 2

[5] L. Beggel, M. Pfeiffer, and B. Bischl. Robust anomaly detection in images using adversarial autoencoders. *Machine Learning and Knowledge Discovery in Databases*, 2019. 3, 4

[6] A. Berg, J. Ahlberg, and M. Felsberg. Unsupervised learning of anomaly detection from contaminated image data using simultaneous encoder training. *arXiv preprint arXiv:1905.11034*, 2019. 3

[7] L. Bergman and Y. Hoshen. Classification-based anomaly detection for general data. In *International Conference on Learning Representations*, 2019. 1, 2, 3, 6, 10, 14

[8] P. Bergmann, M. Fauser, D. Sattlegger, and C. Steger. MVTec AD–a comprehensive real-world dataset for unsupervised anomaly detection. In *CVPR*, 2019. 2, 6

[9] D. Berthelot, N. Carlini, I. Goodfellow, N. Papernot, A. Oliver, and C. A. Raffel. Mixmatch: A holistic approach to semi-supervised learning. *Advances in Neural Information Processing Systems*, 32, 2019. 3

[10] G. Blanchard, G. Lee, and C. Scott. Semi-supervised novelty detection. *JMLR*, 11:2973–3009, 2010. 1

[11] A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. In *CLT*, 1998. 3

[12] P. Branco, L. Torgo, and R. Ribeiro. A survey of predictive modelling under imbalanced distributions. *arXiv preprint arXiv:1505.01658*, 2015. 2

[13] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander. Lof: identifying density-based local outliers. In *International conference on management of data*, 2000. 3

[14] C. E. Brodley, M. A. Friedl, et al. Identifying and eliminating mislabeled training instances. In *Proc National Conference on Artificial Intelligence*, 1996. 3

[15] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357, 2002. 1, 2

[16] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, 2020. 3

[17] F. Denis. PAC learning from positive statistical queries. In *ICALT*, 1998. 1

[18] J. Elson, J. R. Douceur, J. Howell, and J. Saul. Asirra: a captcha that exploits interest-aligned manual image categorization. In *Proc Computer and Communications Security*, 2007. 2, 6

[19] A. Estabrooks, T. Jo, and N. Japkowicz. A multiple resampling method for learning from imbalanced data sets. *Computational intelligence*, 20(1):18–36, 2004. 1, 2

[20] J.-C. Feng, F.-T. Hong, and W.-S. Zheng. Mist: Multiple instance self-training framework for video anomaly detection. In *CVPR*, 2021. 3

[21] M. Galar, A. Fernandez, E. Barrenechea, H. Bustince, and F. Herrera. A review on ensembles for the class imbalance problem: bagging-, boosting-, and hybrid-based approaches. *IEEE Trans on Systems, Man, and Cybernetics*, 42(4):463–484, 2011. 2

[22] I. Golan and R. El-Yaniv. Deep anomaly detection using geometric transformations. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 9781–9791, 2018. 1, 2, 3, 4, 7, 13, 14

[23] N. Görnitz, M. Kloft, K. Rieck, and U. Brefeld. Toward supervised anomaly detection. *Journal of Artificial Intelligence Research*, 46:235–262, 2013. 1

[24] B. Han, Q. Yao, X. Yu, G. Niu, M. Xu, W. Hu, I. W. Tsang, and M. Sugiyama. Co-teaching: robust training of deep neural networks with extremely noisy labels. In *NIPS*, 2018. 3

[25] D. Hendrycks, M. Mazeika, and T. Dietterich. Deep anomaly detection with outlier exposure. In *International Conference on Learning Representations*, 2018. 1, 2, 3, 14

[26] J. P. Hwang, S. Park, and E. Kim. A new weighted approach to imbalanced data classification problem via support vector machine with quadratic cost function. *Expert Systems with Applications*, 38(7):8580–8585, 2011. 1, 2

[27] N. Komodakis and S. Gidaris. Unsupervised representation learning by predicting image rotations. In *International Conference on Learning Representations (ICLR)*, 2018. 3, 7, 13, 14

[28] A. Krizhevsky and G. Hinton. Learning multiple layers of features from tiny images. 2009. 2, 6

[29] C.-H. Lai, D. Zou, and G. Lerman. Robust subspace recovery layer for unsupervised anomaly detection. In *International Conference on Learning Representations*, 2019. 3

[30] L. J. Latecki, A. Lazarevic, and D. Pokrajac. Outlier detection with kernel density functions. In *International Workshop on Machine Learning and Data Mining in Pattern Recognition*, 2007. 2

[31] S. S. Lee. Noisy replication in skewed binary classification. *Computational statistics & data analysis*, 34(2):165–191, 2000. 1, 2

[32] C.-L. Li, K. Sohn, J. Yoon, and T. Pfister. Cutpaste: Self-supervised learning for anomaly detection and localization. In *CVPR*, 2021. 1, 2, 3, 4, 5, 7, 8, 16, 18

[33] J. Li, R. Socher, and S. C. Hoi. Dividemix: Learning with noisy labels as semi-supervised learning. In *ICLR*, 2019. 3

[34] A. Liu, J. Ghosh, and C. E. Martin. Generative oversampling for mining imbalanced datasets. In *DMIN*, pages 66–72, 2007. 1, 2

[35] F. T. Liu, K. M. Ting, and Z.-H. Zhou. Isolation forest. In *ICDM*, 2008. 2, 3

[36] G. J. McLachlan. Iterative reclassification procedure for constructing an asymptotically optimal rule of allocation in discriminant analysis. *Journal of the American Statistical Association*, 70(350):365–369, 1975. 3

[37] X. Meng, S. Wang, Z. Liang, D. Yao, J. Zhou, and Y. Zhang. Semi-supervised anomaly detection in dynamic communication networks. *Information Sciences*, 571:527–542, 2021. 3

[38] 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*, 2018. 3

[39] M. Mohseni, J. Yap, W. Yolland, A. Koochek, and S. Atkins. Can self-training identify suspicious ugly duckling lesions? In *CVPR*, 2021. 3

[40] J. Mũnoz-Marí, F. Bovolo, L. Gómez-Chova, L. Bruzzone, and G. Camp-Valls. Semisupervised one-class support vector machines for classification of remote sensing data. *IEEE transactions on geoscience and remote sensing*, 48(8):3188–3197, 2010. 3

[41] G. Pang, C. Yan, C. Shen, A. v. d. Hengel, and X. Bai. Self-trained deep ordinal regression for end-to-end video anomaly detection. In *CVPR*, 2020. 3, 4

[42] D. A. Reynolds. Gaussian mixture models. *Encyclopedia of biometrics*, 741:659–663, 2009. 2

[43] L. Ruff, R. Vandermeulen, N. Goernitz, L. Deecke, S. A. Siddiqui, A. Binder, E. Müller, and M. Kloft. Deep one-class classification. In *ICML*, 2018. 1, 2, 3, 4

[44] L. Ruff, R. A. Vandermeulen, N. Görnitz, A. Binder, E. Müller, K.-R. Müller, and M. Kloft. Deep semi-supervised anomaly detection. In *ICLR*, 2020. 1, 3

[45] B. Schölkopf, R. C. Williamson, A. J. Smola, J. Shawe-Taylor, J. C. Platt, et al. Support vector method for novelty detection. In *NIPS*, 1999. 1, 2

[46] H. Scudder. Probability of error of some adaptive pattern-recognition machines. *IEEE Transactions on Information Theory*, 11(3):363–371, 1965. 3

[47] M. Sezgin and B. Sankur. Survey over image thresholding techniques and quantitative performance evaluation. *Journal of Electronic imaging*, 13(1):146–165, 2004. 5, 18

[48] K. Sohn, D. Berthelot, N. Carlini, Z. Zhang, H. Zhang, C. A. Raffel, E. D. Cubuk, A. Kurakin, and C.-L. Li. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *NeurIPS*, 2020. 3

[49] K. Sohn, C.-L. Li, J. Yoon, M. Jin, and T. Pfister. Learning and evaluating representations for deep one-class classification. In *ICLR*, 2021. 1, 2, 3, 4, 5, 6, 7, 8, 13, 14, 15, 17

[50] H. Song, Z. Jiang, A. Men, and B. Yang. A hybrid semi-supervised anomaly detection model for high-dimensional data. *Computational intelligence and neuroscience*, 2017. 3

[51] D. M. Tax and R. P. Duin. Support vector data description. *Machine learning*, 54(1):45–66, 2004. 1, 2

[52] Y. Xia, X. Cao, F. Wen, G. Hua, and J. Sun. Learning discriminative reconstructions for unsupervised outlier removal. In *ICCV*, 2015. 3

[53] H. Xiao, K. Rasul, and R. Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017. 6

[54] Q. Xie, M.-T. Luong, E. Hovy, and Q. V. Le. Self-training with noisy student improves imagenet classification. In *CVPR*, 2020. 3

[55] B. Zhang and W. Zuo. Learning from positive and unlabeled examples: A survey. In *2008 International Symposiums on Information Processing*, 2008. 1

[56] C. Zhou and R. C. Paffenroth. Anomaly detection with robust deep autoencoders. In *KDD*, 2017. 2, 3, 4

[57] B. Zong, Q. Song, M. R. Min, W. Cheng, C. Lumezanu, D. Cho, and H. Chen. Deep autoencoding gaussian mixture model for unsupervised anomaly detection. In *ICLR*, 2018. 1, 2, 3, 6

A Appendix

A.1 Additional results with different metrics

There are various performance metrics of OCCs. In the main manuscript, we mainly use AUC, Average Precision (AP) and F1-score as the evaluation metrics. In this subsection, we report the performances of the proposed model (SRR) and baselines in terms on Recall at Precision 70 and 90 as the additional metrics on CIFAR-10 dataset.

As in Fig. 9, we observe a similar trends with these two additional metrics as well. For example, the performances of SRR are robust across various anomaly ratios. On the other hand, all the other

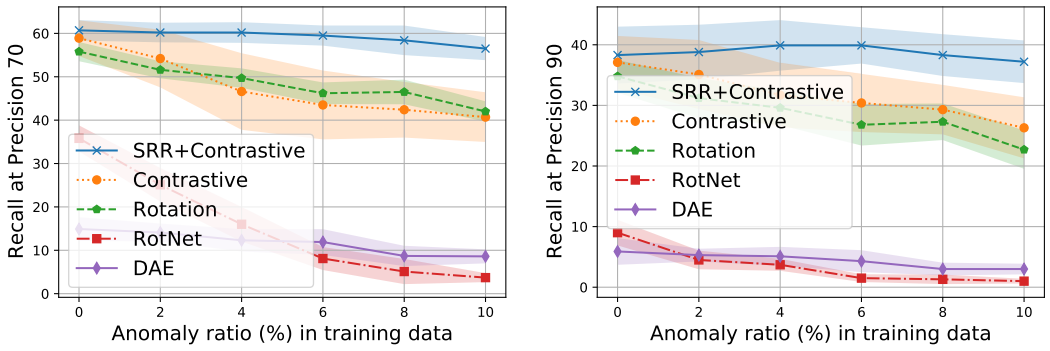
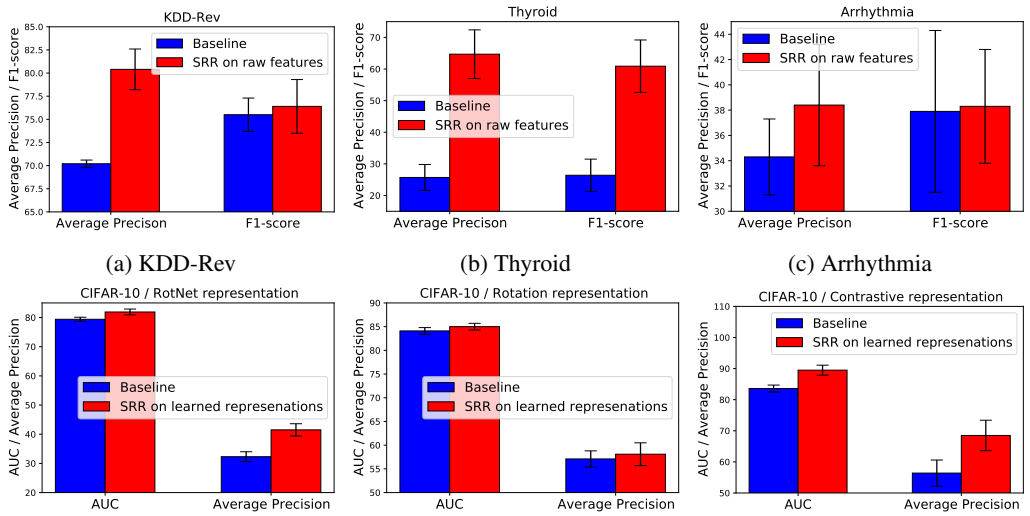


Figure 9: Performance of various OCCs on CIFAR-10 dataset. SRR is applied on top of Contrastive [49]. (Left) Recall at Precision 70, (Right) Recall at Precision 90.

OCCs show consistent and significant performance degradation as the anomaly ratio increases. SRR performs 15.8 and 10.9 better than the state-of-the-art OCC [49] in terms of recall at precision 70 and 90, respectively.

A.2 SRR ON RAW TABULAR FEATURES / LEARNED IMAGE REPRESENTATIONS



(a) KDD-Rev (b) Thyroid (c) Arrhythmia

(d) RotNet [22] (e) Rotation [27] (f) Contrastive [49]

Figure 10: Performance of SRR on (top) raw tabular features and (lower) learned image representations. SRR consistently outperforms baseline and in some cases (e.g., Thyroid, and Contrastive [49]), the performance improvements are significant.

SRR is also applicable on raw tabular features or learned image representation without representation update using data refinement. In this section, we demonstrate the performance improvements by SRR without representation update to verify the effectiveness of data refinement block of SRR for shallow OCCs.

Fig. 10 (upper) demonstrates consistent and significant performance improvements when we apply SRR on top of raw tabular features. Specifically, the Average Precision (AP) improvements are 10.2, 29.0, and 4.1 with KDD-Rev, Thyroid, and Arrhythmia tabular datasets, respectively.

We also apply SRR on top of various learned image representations. As can be seen in Fig. 10 (lower), the performance improvements of SRR are consistent across various different learned image representations (without representation update). For instance, the AP improvements are 9.2, 1.0,

and 12.1 with learned image representations using RotNet [22], Rotation [27], and Contrastive [49], respectively.

A.3 Implementation Details on GOAD [7] for Tabular Data Experiments

A classification-based AD method, GOAD [7], has demonstrated strong AD performance on tabular datasets. Unlike previous works [22, 25] that formulate a parametric classifier for multiple transformation classification, GOAD employs distance-based classification of multiple transformations. For the set of transformations $T_m : \mathcal{X} \to \mathcal{D}$, $m = 1, ..., M$, the loss function of GOAD is written as in equation 3 with the probability defined in equation 4.

$$\mathcal{L} = -\mathbb{E}_{m,x}\big[\log P(m|T_m(x))\big], \tag{3}$$

$$P(\hat{m}|T_m(x)) = \frac{\exp(-\|f(T_m(x)) - c_{\hat{m}}\|^2)}{\sum_n \exp(-\|f(T_m(x)) - c_n\|^2)}, \tag{4}$$

where the centers c_m's are updated by the average feature over the training set. While it is shown to perform well [7], we find that the distance-based formulation is not necessary, and we achieve the similar performance, if not worse, to [7] using a parametric classifier when computing the probability:

$$P(\hat{m}|T_m(x)) = \frac{\exp\left(w_{\hat{m}}^\top f(T_m(x)) + b_{\hat{m}}\right)}{\sum_n \exp\left(w_n^\top f(T_m(x)) + b_n\right)} \tag{5}$$

The formulation in equation 5 is easier to optimize than its original form in equation 4 as it can be fully optimized with backpropagation without alternating updates of feature extractor f and centers c_m. Once we learn a representation by optimizing the loss in equation 3 using equation 5, we follow a two-stage one-class classification framework of [49] to construct a set of Gaussian density estimation OCCs for each transformation. Finally, we aggregate a maximum normality scores from a set of classifiers as the normality score.

In Table 1, we summarize the implementation details, such as network architecture or hyperparameters, and AD performance under N setting that reproduces the results in [7].

Datasets	KDD	KDD-Rev	Thyroid	Arrhythmia
F-score [7]	98.4±0.2	98.9±0.3	74.5±1.1	52.0±2.3
F-score (ours)	98.0±0.2	95.0±0.2	75.1±2.4	54.8±3.2
f (feature)	[Linear(8), LeakyReLU(0.2)] ×5			
Optimizer	Momentum SGD (momentum= 0.9)			
Learning rate	0.001			
L2 weight regularization	0.00003			
Batch size	$64 \times M$			
Random projection dimension	32			
M (number of transformations)	32		256	
Train steps	2^{10}		2^{16}	

Table 1: The AD performance under N setting of GOAD in [7] and our implementation. Our implementation demonstrates comparable, if not worse, performance to those reported in [7]. Our implementation also shares most hyperparameters across datasets except the M, the number of transformations, and the train steps, which are closely related to the size of training data.

A.4 PER-CLASS AUC AND AP

In the main manuscript, we report the mean and standard deviation of AUC and AP across all classes in each dataset. In this section, we report the mean and standard deviation of AUC and AP for each class in each dataset (including CIFAR-10 (Table 2), MVTec (Table 3), fMNIST (Table 4), and Dog-vs-Cat (Table 5) datasets).

A.4.1 PER-CLASS AUC AND AP ON CIFAR-10 DATASET

anomaly ratios	0%		10%	
Classes / Methods	Baseline [49]	SRR + [49]	Baseline [49]	SRR + [49]
AUC				
Airplane	90.5±0.6	90.4±0.4	85.3±0.8	87.6±0.7
Automobile	98.8±0.1	98.7±0.1	95.2±0.5	97.9±0.3
Bird	87.5±0.6	87.1±0.8	75.6±1.1	84.7±1.2
Cat	81.5±0.5	80.3±0.7	68.9±1.0	78.5±1.5
Deer	90.3±0.4	90.2±0.5	79.9±0.8	87.4±0.5
Dog	90.8±0.7	90.8±0.6	79.2±2.5	89.6±0.6
Frog	92.0±0.6	91.4±0.9	78.5±2.1	87.5±1.3
Horse	97.8±0.1	97.9±0.1	93.3±0.5	96.0±0.6
Ship	96.5±0.2	96.4±0.1	91.5±0.9	95.2±0.5
Truck	95.1±0.3	95.8±0.1	88.9±0.4	94.7±0.4
Average Precision				
Airplane	68.3±1.7	67.9±1.1	61.3±1.9	61.8±1.1
Automobile	93.9±0.2	94.1±0.1	85.9±3.4	91.5±0.7
Bird	62.2±2.1	61.7±2.4	42.4±3.0	57.5±2.3
Cat	44.0±2.4	42.9±1.9	30.3±3.8	44.9±0.9
Deer	59.5±0.8	58.8±1.2	42.9±3.5	53.6±1.5
Dog	59.1±6.5	59.8±4.7	39.7±9.1	64.5±2.0
Frog	72.7±2.0	70.4±6.9	38.5±11.0	73.6±3.6
Horse	89.0±0.7	89.2±0.6	80.6±1.9	86.2±1.1
Ship	83.9±0.8	83.7±0.6	74.9±2.4	79.8±0.7
Truck	79.0±1.6	81.3±0.4	67.0±1.9	75.8±1.2

Table 2: The AD performance per class on CIFAR-10 dataset in terms of AUC and AP. Class represents the label of normal samples.

A.4.2 PER-CLASS AUC AND AP ON MVTEC DATASET

anomaly ratios	0%		10%	
Classes / Methods	Baseline [32]	SRR + [32]	Baseline [32]	SRR + [32]
AUC				
Bottle	98.0$\pm$1.0	98.8$\pm$0.7	94.3$\pm$0.8	98.7$\pm$0.7
Cable	77.7$\pm$0.7	77.9$\pm$2.6	72.9$\pm$3.3	80.4$\pm$0.9
Capsule	96.1$\pm$1.1	97.2$\pm$1.3	92.4$\pm$1.7	96.8$\pm$0.7
Hazelnut	97.2$\pm$0.7	97.5$\pm$0.4	83.5$\pm$3.6	96.7$\pm$0.6
Metal Nut	98.5$\pm$1.0	98.6$\pm$0.4	91.2$\pm$2.8	98.7$\pm$0.3
Pill	92.1$\pm$3.1	91.1$\pm$2.5	85.1$\pm$1.5	92.1$\pm$0.5
Screw	87.4$\pm$1.6	86.8$\pm$2.5	79.9$\pm$1.0	86.4$\pm$1.4
Toothbrush	98.5$\pm$1.5	98.3$\pm$1.1	98.6$\pm$1.3	99.3$\pm$1.2
Transistor	96.4$\pm$2.0	96.7$\pm$1.6	93.9$\pm$2.8	95.7$\pm$0.7
Zipper	99.4$\pm$0.4	98.0$\pm$0.6	91.1$\pm$2.9	99.2$\pm$0.7
Carpet	90.0$\pm$3.9	90.8$\pm$4.6	80.4$\pm$6.1	91.2$\pm$2.7
Grid	99.2$\pm$0.5	99.7$\pm$0.3	92.9$\pm$7.6	99.7$\pm$0.2
Leather	99.8$\pm$0.3	99.7$\pm$0.4	98.0$\pm$1.1	100.0$\pm$0.0
Tile	93.1$\pm$1.5	93.3$\pm$2.7	86.4$\pm$3.4	92.4$\pm$1.8
Wood	98.8$\pm$0.9	98.5$\pm$1.4	92.1$\pm$2.2	99.0$\pm$0.5
Average Precision				
Bottle	98.8$\pm$0.7	99.3$\pm$0.5	94.3$\pm$1.6	98.0$\pm$0.6
Cable	81.8$\pm$0.9	82.3$\pm$1.7	70.8$\pm$4.3	74.1$\pm$3.0
Capsule	99.0$\pm$0.3	99.3$\pm$0.3	79.5$\pm$5.3	89.8$\pm$2.7
Hazelnut	97.1$\pm$0.6	97.0$\pm$0.4	88.8$\pm$3.6	93.6$\pm$1.9
Metal Nut	99.6$\pm$0.3	99.6$\pm$0.1	76.1$\pm$8.4	92.1$\pm$3.1
Pill	98.2$\pm$0.7	97.9$\pm$0.6	56.6$\pm$2.6	67.3$\pm$2.2
Screw	93.8$\pm$1.1	93.6$\pm$1.3	60.1$\pm$2.8	63.7$\pm$1.8
Toothbrush	99.2$\pm$0.9	99.2$\pm$0.6	97.4$\pm$2.2	98.6$\pm$2.0
Transistor	93.0$\pm$3.0	93.7$\pm$2.4	97.8$\pm$1.6	97.1$\pm$0.5
Zipper	99.8$\pm$0.2	99.2$\pm$0.3	86.1$\pm$2.8	98.0$\pm$1.5
Carpet	94.4$\pm$3.5	95.7$\pm$2.8	64.2$\pm$8.2	71.6$\pm$4.2
Grid	99.5$\pm$0.3	99.8$\pm$0.2	90.9$\pm$6.1	99.4$\pm$0.4
Leather	99.9$\pm$0.2	99.9$\pm$0.2	97.2$\pm$1.3	100.0$\pm$0.1
Tile	96.1$\pm$0.9	96.4$\pm$1.4	82.4$\pm$4.5	86.4$\pm$2.8
Wood	99.4$\pm$0.5	99.4$\pm$0.6	88.1$\pm$3.2	97.2$\pm$1.7

Table 3: The AD performance per class on MVTec dataset in terms of AUC and AP. Class represents the label of normal samples.

A.4.3 PER-CLASS AUC AND AP ON FMNIST DATASET

anomaly ratios	0%		10%	
Classes / Methods	Baseline [49]	SRR + [49]	Baseline [49]	SRR + [49]
AUC				
T-shirt/top	93.3±1.2	93.2±1.0	79.2±1.3	84.7±1.0
Trouser	99.2±0.3	99.2±0.2	92.9±0.8	94.5±1.9
Pullover	93.8±0.2	93.8±0.2	78.4±2.7	90.1±1.1
Dress	94.4±0.5	94.0±0.5	78.4±3.5	85.8±1.6
Coat	95.1±0.1	94.9±0.2	80.4±3.1	88.7±1.7
Sandal	95.4±0.3	95.5±0.4	67.6±4.3	81.7±1.8
Shirt	87.9±0.2	87.9±0.2	73.9±1.2	81.0±2.5
Sneaker	99.1±0.0	99.2±0.1	80.2±4.9	94.7±3.6
Bag	97.8±0.2	97.6±0.3	58.2±3.3	68.6±2.3
Ankle boot	98.6±0.2	98.5±0.2	76.6±2.6	87.1±3.6
Average Precision				
T-shirt/top	68.7±12.1	67.9±8.9	43.0±4.7	56.8±3.7
Trouser	96.5±2.9	98.2±0.3	78.2±5.2	87.1±3.7
Pullover	62.7±1.2	63.6±1.3	27.2±6.9	57.1±1.5
Dress	72.3±3.7	71.6±2.7	36.1±8.8	57.3±2.6
Coat	66.2±1.1	65.6±1.0	24.3±3.9	50.7±9.5
Sandal	70.9±2.8	70.8±2.4	17.8±5.9	40.1±5.4
Shirt	49.6±0.9	49.7±0.9	27.2±5.8	37.6±4.7
Sneaker	91.9±0.4	92.9±0.9	21.0±3.8	81.6±11.4
Bag	92.2±0.4	91.8±1.0	13.2±2.0	20.1±3.7
Ankle boot	87.9±1.4	86.5±1.5	19.6±3.1	48.0±16.5

Table 4: The AD performance per class on fMNIST dataset in terms of AUC and AP. Class represents the label of normal samples.

A.4.4 PER-CLASS AUC AND AP ON DOG-VS-CAT DATASET

anomaly ratios	0%		10%	
Classes / Methods	Baseline [49]	SRR + [49]	Baseline [49]	SRR + [49]
AUC				
Dog	89.7±0.6	89.2±0.4	74.7±2.1	79.1±1.3
Cat	88.2±0.4	88.0±0.7	70.1±2.4	70.7±1.8
Average Precision				
Dog	91.6±0.7	91.0±0.6	80.0±2.2	84.0±1.2
Cat	90.2±0.3	89.8±1.0	71.6±3.8	71.5±4.8

Table 5: The AD performance per class on Dog-vs-Cat dataset in terms of AUC and AP. Class represents the label of normal samples.

A.5 SRR WITH OTSU'S METHOD

In the main manuscript we show that the performances of the proposed unsupervised anomaly detection framework (SRR) are significantly better than the state-of-the-art OCCs if the given data is fully unlabeled. Although SRR does not need any labeled data, it needs a true anomaly ratio for hyper-parameter optimization. In this subsection, we further extend SRR to the setting where there is no access to the true anomaly ratio as well. We utilize the Otsu's method [47] to identify the threshold between normal and anomalous samples and use that as the hyper-parameter (γ) instead of twice of the true anomaly ratio.

The key idea of the Otsu's method is to find the threshold that minimizes the intra-class variance. This is defined as the weighted sum of variances of the two classes. Let us denote the normality scores as $\{s_i\}_{i=1}^N$ and threshold as η. Then, we try to find the threshold (η) that minimizes the weighted sum of the variance $(w_0(\eta) \times \sigma_0(\eta) + w_1(\eta) \times \sigma_1(\eta))$ where $w_0(\eta) = \sum_{i=1}^N \mathbb{I}(s_i < \eta)/N$ and $w_1(\eta) = \sum_{i=1}^N \mathbb{I}(s_i \geq \eta)/N$. $\sigma_0(\eta)$ and $\sigma_1(\eta)$ are the variances of each class. The optimal threshold (η^*) is determined as

$$\eta^* = \min_{\eta} w_0(\eta) \times \sigma_0(\eta) + w_1(\eta) \times \sigma_1(\eta).$$

We use the twice of η^* as the hyperparameter (γ) in SRR.

We evaluate the performances of Otsu's method (on top of SRR) in comparison to the state-of-the-art OCC [32] and original SRR with the knowledge of the true anomaly ratio using MVTec dataset.

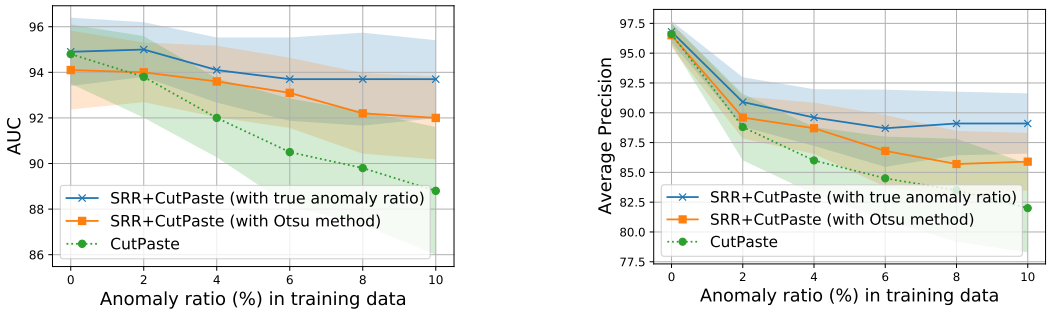


Figure 11: Unsupervised anomaly detection performances with Otsu's method on top of SRR on MVTec dataset. (Left) AUC, (Right) Average Precision (AP).

Fig. 11 demonstrates that even without true anomaly ratio, the performance of SRR can be significantly better than the state-of-the-art OCC [32] with Otsu's method. Also, Fig. 11 shows that the knowledge of true anomaly ratio is crucial information for maximizing the performance of SRR in fully unsupervised settings.