Set Features for Fine-grained Anomaly Detection

Niv Cohen, Issar Tzachor, Yedid Hoshen

Set Features for Fine-grained Anomaly Detection

Niv Cohen **Issar Tzachor** **Yedid Hoshen**
School of Computer Science and Engineering
The Hebrew University of Jerusalem, Israel
nivc@cs.huji.ac.il

Abstract

Fine-grained anomaly detection has recently been dominated by segmentation-based approaches. These approaches first classify each element of the sample (e.g., image patch) as normal or anomalous and then classify the entire sample as anomalous if it contains anomalous elements. However, such approaches do not extend to scenarios where the anomalies are expressed by an unusual combination of normal elements. In this paper, we overcome this limitation by proposing set features that model each sample by the distribution of its elements. We compute the anomaly score of each sample using a simple density estimation method. Our simple-to-implement approach[1] outperforms the state-of-the-art in image-level logical anomaly detection ($+3.4\%$) and sequence-level time series anomaly detection ($+2.4\%$).

1 Introduction

Anomaly detection aims to automatically identify samples that exhibit unexpected behavior. In fine-grained anomaly detection, such as detecting faults in industrial images or irregularities in time series, anomalies are quite subtle. For example, let us consider an image of a bag containing screws, nuts and washers (Fig.1). There are two ways in which a sample can be anomalous: (i) one or more of the elements in the sample are anomalous. E.g., a broken screw. (ii) the elements are normal but appear in an anomalous combination. E.g., one of the washers might be replaced with a nut.

In recent years, remarkable progress was made in detecting samples featuring anomalous elements. The usual procedure is: First, we perform anomaly segmentation by detecting which (if any) of the elements of the sample are anomalous. This can be performed by a variety of methods, in particular, using density estimation methods Cohen & Hoshen (2020); Defard et al. (2021); Roth et al. (2022). Given an anomaly segmentation map, we compute the sample-wise anomaly score as the number of anomalous elements, or the abnormality level of the most anomalous element. If the anomaly score exceeds a threshold, the entire sample is denoted as an anomaly. We denote this paradigm: detection-by-segmentation.

Here, we tackle the more challenging case of detecting anomalies consisting of an unusual combination of normal elements. For example, consider the case where normal images contain two washers and two nuts but anomalous images may contain one washer and three nuts. As each of the elements (nuts or screws) occur in natural images, simple detection-by-segmentation will not work. Instead, a more holistic understanding of the image is required. While simple global representations, e.g., taking the average of the representations of all elements, might work in some cases, the result is typically too coarse to detect fine-grained anomalies.

We propose to detect anomalies consisting of unusual combinations of normal elements using set representations. The key assumption behind our method is that the distribution of elements in a sample is more correlated with it being anomalous than the ordering of the elements. Each sample

[1]The code is available on github under: https://github.com/NivC/SINBAD.

.

is therefore modeled as an orderless set of elements. The elements are represented using feature embedding, e.g., a deep representation extracted by a pre-trained neural network or handcrafted features. To describe this set of features we count the percentage of elements falling in different histograms bins. We compute a histogram for each dimension of the feature space. The bins from all the histograms are concatenated together, forming our set representation. Finally, we score anomalies using density estimation on this set representation.

Our method, *SINBAD* (*Set IN*spection *B*ased *A*omalies *D*etection) is evaluated on two diverse tasks. The first task is image-level logical anomaly detection on the MVTec-LOCO datasets. Our method outperforms more complex state-of-the-art methods, while not requiring any training. We also evaluate our method on series-level time series anomaly detection. Our approach outperforms all current methods while not using augmentation or training.

We make the following contribution:

- Identifying set representation as key for detecting anomalies consisting entirely of normal elements.
- An effective approach for measuring the distance between samples treated as sets of their local elements.
- State-of-the-art results for sample-wise logical anomaly detection (MVTec-LOCO) and time series datasets.

2 Previous work

Image Anomaly Detection. The field of anomaly detection has been researched for several decades. A comprehensive review can be found in Ruff et al. (2021). Early approaches (Glodek et al. (2013); Latecki et al. (2007); Eskin et al. (2002)) used handcrafted representations and aimed at detecting images with different coarse-grained objects (e.g., cats vs. dogs). Deep learning has provided a significant improvement on such benchmarks Larsson et al. (2016); Ruff et al. (2018); Golan & El-Yaniv (2018); Hendrycks et al. (2019); Ruff et al. (2019); Perera & Patel (2019); Salehi et al. (2021); Tack et al. (2020). As density estimation methods utilizing pre-trained deep representation have made significant steps towards the supervised performance on such benchmarks Deecke et al. (2021); Cohen & Avidan (2022); Reiss et al. (2021); Reiss & Hoshen (2021); Reiss et al. (2022), much research is now directed at other challenges Reiss et al. (2022). Such challenges include detecting anomalous image parts which are small and fine-grained Cohen & Hoshen (2020); Li et al. (2021); Defard et al. (2021); Roth et al. (2022); Horwitz & Hoshen (2022). The recent progress in anomaly detection and segmentation methods for this setting has been enabled by the introduction of appropriate datasets Bergmann et al. (2019, 2021); Carrera et al. (2016); Jezek et al. (2021). The dominant paradigm in state-of-the-art methods Roth et al. (2022) is to detect fine-grained anomalous images by first segmenting highly anomalous patches and then scoring the entire image based on these segmentation maps. Recently, the MVTec-LOCO datasetBergmann et al. (2022) has put the spotlight on fine-grained anomalies which cannot be identified using single patches, but only when examining the connection between different (otherwise normal) elements in an image. Here, we will focus on detecting such *logical* anomalies.

Time series Anomaly detection. The task of anomaly detection in time series has been studied over several decades (Blázquez-García et al., 2021). In this paper, we are concerned with anomaly detection of entire sequences, i.e. cases where an entire signal may be abnormal. Traditional approaches for this task include generic anomaly detection approaches such as k nearest neighbors (kNN) based methods e.g. vanilla kNN (Eskin et al., 2002) and Local Outlier Factor (LOF) (Breunig et al., 2000), Tree-based methods (Liu et al., 2008), One-class classification methods (Tax & Duin, 2004) and SVDD (Schölkopf et al.). Some traditional methods such as auto-regressive methods are particular to time series anomaly detection (Rousseeuw & Leroy, 2005). With the advent of deep learning, the traditional approaches were augmented with deep-learned features. Deep one-class classification methods include DeepSVDD (Ruff et al., 2018) and DROCC (Goyal et al., 2020). Deep auto-regressive methods include RNN-based prediction and auto-encoding methods (Bontemps et al., 2016; Malhotra et al., 2016). In addition, some deep learning anomaly detection approaches were proposed that are conceptually different from traditional approaches. These methods are based on the premise that classifiers trained on normal data will struggle to generalize to anomalous data. These

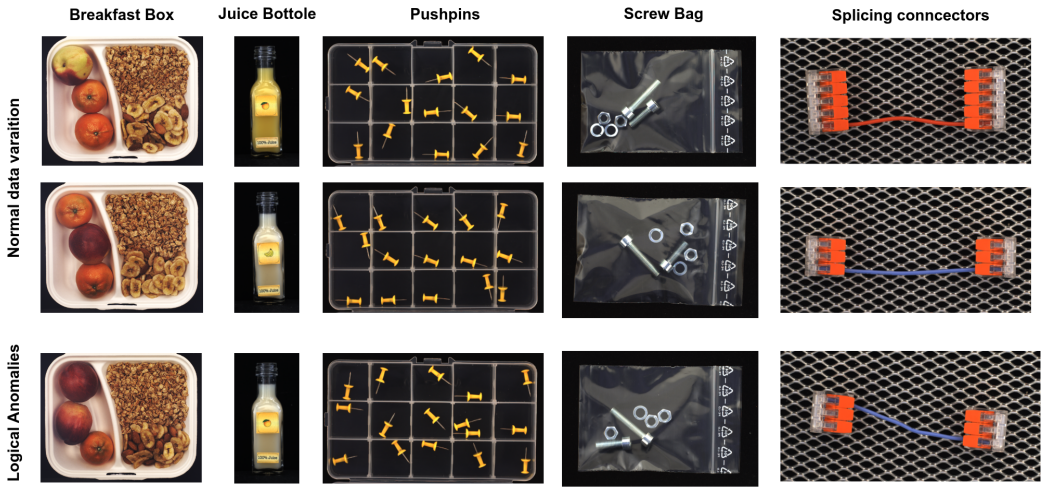

Figure 1: In logical anomalies, each image element (e.g., patch) may be normal even when their combination is anomalous. These cases are challenging as the variation among the normal data can be large while anomalies are fine-grained (e.g., swapping a bolt and a washer in the *screw bag* class).

approaches were originally developed for image anomaly detection (Golan & El-Yaniv, 2018) but have been extended to tabular and time series data (Bergman & Hoshen, 2020; Qiu et al., 2021).

Discretized Projections. Discretized projections of multivariate data have been used in many previous works. Locally sensitive hashing Dasgupta et al. (2011) uses random projection and subsequent binary quantization as a hash for high-dimensional data. It was used to facilitate fast k nearest neighbor search. Random projections transformation is also highly related to the Radon transform Radon (1917). Kolouri et al. (Kolouri et al., 2015) used this representation as a building block in their set representation. HBOS Goldstein & Dengel (2012) performs anomaly detection by representing each dimension of multivariate data using a histogram of discretized variables and subsequent density estimation. LODA Pevnỳ (2016) extends this work, by first projecting the data using a random projection matrix (followed by discretization). We differ from LODA in the use of a different density estimator and in using sets of multiple elements rather than single sample descriptions. Rocket and mini-rocket Dempster et al. (2020, 2021) represent time series for classification using the averages of their window projection.

3 Set Features for Anomaly Detection

3.1 A Set is More Than the Sum of Its Parts

Detecting logical image anomalies, or collective time series anomalies, requires understanding how the different parts of each sample interact with one another. As a motivating example let us consider the *screw bag* class from the MVTec-LOCO dataset (Fig. 1). Each normal sample in this class contains two screws (of different lengths), two nuts, and two washers. Anomalies may occur for example when one screw is missing, or when an additional nut replaces one of the washers. Detecting anomalies such as these requires a joint description of all elements within the sample since each local element could have come from a normal sample.

The typical way to aggregate element descriptor features is by average pooling. However, this is not always suitable for set anomaly detection. In supervised learning, average pooling is often built into architectures such as ResNet He et al. (2016) or DeepSets Zaheer et al. (2017), in order to aggregate local features. Therefore, deep features learnt with a supervised loss are already trained to be effective for pooling. However, for lower-level feature descriptors, such as the ones we use here, this may not be the case. As demonstrated in Fig.2, the average of a set of features is far from a complete description of the set. This is especially true in anomaly detection, where density estimation approaches require more discriminative features than those needed for supervised learning Reiss et al.

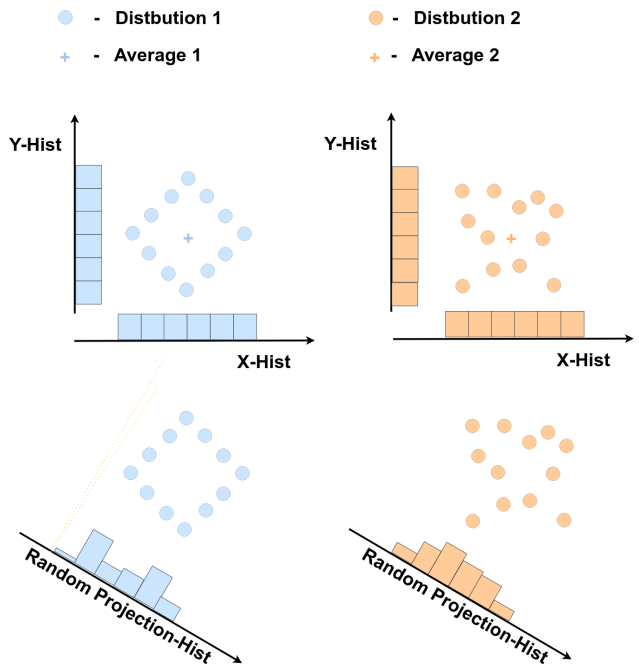

Figure 2: Random projection histograms allow us to distinguish between sets where other methods could not. The two sets are similar in their averages and histograms along the original axes, but result in different histograms when projected along a random axis.

(2022). This means that even when an average pooled set of features worked for a supervised task, it might not work for anomaly detection.

Here, we wish to describe a set by modeling the distribution of its elements, ignoring the ordering between them. A naive way of doing so is by a discretized, volumetric representation, similarly to 3D voxels for point clouds. Unfortunately, such approaches cannot scale to high dimensions, and more compact representations are required. In this work, we choose to represent sets using a collection of 1D histograms. Each histogram represents the density of the elements of the set when projected along a particular direction. We provide an illustration of this idea in Figure 2.

Projecting a set along its original axes may not be discriminative enough. Histograms along the original axes correspond to 1D marginals, and may map distant elements to the same histogram bins (see 2 for an illustration). On the other side, we can see at the bottom of the figure that when the set elements are first projected along another direction, the histograms of the two sets are distinct. This suggests a method for set description: first project each set along a shared random direction and then compute a 1D histogram for each set along this direction. We can obtain a more powerful descriptor by repeating this procedure with projections along multiple random directions.

3.2 Preliminaries

We are provided a training set $\mathcal{S}$ containing a set of N_S samples $x_1, x_2..x_{N_S} \in \mathcal{S}$. All the samples at training time are known to be normal. At test time, we are presented with a new sample $\tilde{x}$. Our objective is to learn a model, which operates on each sample $\tilde{x}$ and outputs an anomaly score. Samples with anomaly scores higher than a predetermined threshold value are labeled as anomalies.

The unique aspect of our method is its treatment of each sample x as consisting of a set of N_E elements $x = [e_1, e_2..e_{N_E}]$. Examples of such elements include patches for images and temporal windows for time series. We assume the existence of a powerful feature extractor F that maps each raw element e into an element feature descriptor f_e. We will describe specific implementations of the feature extraction for two important applications: images and time series, in section 4.

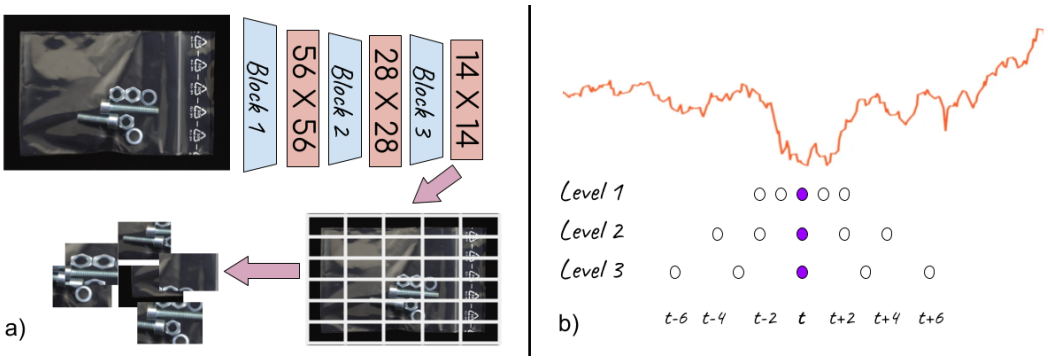

Figure 3: For both image and time series samples we extract set elements of different granularity. In image samples (left), the sets of elements are extracted from different ResNet levels. For time series data (right), we take pyramids of windows at different strides around each time step (noted in blue circles).

3.3 Set Features by Histogram of Projections

Motivated by the toy example in section 3.1, we propose to model each set x by computing a histogram of the values of its elements along each direction. As the given raw axes of the representation may mask out interesting degrees of variation, we perform a random projection prior to building the histograms.

Histogram descriptor. As explained in section 3.1, average pooling the features of all elements in the set may result in insufficiently informative representations. Instead, we describe the set using the histogram of values along each dimension. We note the set of the values of the jth feature in each element of each sample as $s[j] = \{f_1[j], f_2[j]..f_{N_S}[j]\}$. We compute the maximal and minimal values for sets $s[j]$ across all the samples, and divide the region between them into K bins. We compute histograms H_j for each of the N_D dimensions and concatenate them into a single set descriptor h. The descriptor of each set therefore has a dimension of $N_D \cdot K$.

Projection. As discussed before, not all projection directions are equally informative for describing the distributions of sets. In the general case, it is unknown which directions will be the most informative ones for capturing the difference between normal and anomalous sets. As we cannot tell the best projection directions in advance, we propose to randomly project the features. This ensures a low likelihood for catastrophically poor projection directions, such as those in the example in Fig.2.

In practice, we generate a random projection matrix $P \in R^{(N_D, N_P)}$ by sampling values for each dimension from the Gaussian distribution $\mathbb{N}(0, 1)$. We project the features f of each element of x, yielding projected features f':

$$f' = Pf \tag{1}$$

We run the histogram descriptor procedure described above on the projected features. The final set descriptor h_x therefore becomes the concatenation of N_P histograms, resulting in a dimension of $N_P \cdot K$.

3.4 Anomaly scoring

We now wish to use the histogram features h_x to detect anomalous samples. We estimate the probability density function (PDF) of the normal data features $\{h_x\} \in S$ using a Gaussian density estimator. We provide theoretical justification for the distributional assumption in App. F. We compute the mean μ and covariance Σ of the descriptors of all sets. The estimated PDF is given by:

$$p(h) = \mathcal{N}(h|\mu, \Sigma) \tag{2}$$

As the anomaly score is correlated with the negative-likelihood of a sample, we define the anomaly score $a(h)$ as the negative log-likelihood. Ignoring constant terms, this becomes the Mahalanobis distance:

$$a(h) = (h - \mu)^T \Sigma^{-1} (h - \mu) \tag{3}$$

In practice, we found that the Mahalanobis distance to the kNN (kNN with whitening transformation) rather than to μ worked slightly better and is therefore used to compute our results.

4 Application to Image and Time Series Anomaly Detection

In this section, we apply our set description method for anomaly detection in image and time series data.

4.1 Images as Sets

Images can be seen as consisting of a set of elements of different levels of granularity. This ranges from pixels to small patches and low-level elements such as lines or corners, up to high-level elements such as objects. For anomaly detection, we typically do not know in advance the correct level of granularity for separating between normal and anomalous samples. This depends on the anomalies, which are unknown during training. Instead, we perform anomaly detection for different levels of granularity and combine the scores. These levels of granularity correspond to patches of different sizes.

In practice, we use representations from intermediate blocks of a pre-trained ResNet He et al. (2016). As a ResNet network simultaneously embeds many local patches of each image, we pass the image through the network encoder and extract our representations from the intermediate activations at the end of different residual blocks (see Fig.3). We define each spatial location in the activation map as an element. Note that as different blocks have different resolutions, they yield different numbers of elements. We take the elements at the end of each residual block as our sets.

4.2 Time Series as Sets

Time series data can be viewed as a set of temporal windows. Similarly to images, it is generally not known in advance which temporal scale is relevant for detecting anomalies; i.e. the duration of windows which includes the semantic phenomenon. Inspired by *Rocket* Dempster et al. (2020), we define the basic elements of a time series as a collection of temporal window pyramids. Each pyramid contains L windows. All the windows in a pyramid are centered at the same time step, each containing τ samples (Fig.3). The first level window includes τ elements with stride 1, the second level window includes τ elements with stride 2, etc. Such window pyramid is computed for each time step in the time series, and the entire series is represented as the set of its pyramid elements. More implementation details are described in Sec.C.2.

5 Results

5.1 Logical Anomaly Detection Results

Logical Anomalies Dataset. We use the recently published MVTec-LOCO dataset Bergmann et al. (2022) to evaluate our method's ability to detect anomalies caused by unusual configurations of normal elements. This dataset features five different classes: *breakfast box*, *juice bottle*, *pushpins*, *screw bag* and *splicing connector* (see Fig.1). Each class includes: (i) a training set of normal samples exhibiting the normal variation of the class (~ 350 samples). (ii) a validation set, containing another, smaller, set of normal samples (~ 60 samples). (iii) a test set, containing normal samples, structural anomalies, and logical anomalies (~ 100 samples each).

The anomalies in each class are divided into *structural anomalies* and *logical anomalies*. Structural anomalies feature local defects, somewhat similar to previous datasets such as Bergmann et al. (2019). Conversely, logical anomalies may violate 'logical' conditions expected from the normal data. As

Table 1: Anomaly detection on MVTec-LOCO. ROC-AUC (%). See Tab.5 for the full table.

		f-AnoGAN	MNAD	ST	SPADE	PCore	GCAD	SINBAD
Logical Anomalies	Breakfast box	69.4	59.9	68.9	81.8	77.7	87.0	**96.5 ± 0.1**
	Juice bottle	82.4	70.5	82.9	91.9	83.7	**100.0**	96.6 ± 0.1
	Pushpins	59.1	51.7	59.5	60.5	62.2	**97.5**	83.4 ± 3.0
	Screw bag	49.7	60.8	55.5	46.8	55.3	56.0	**78.6 ± 0.1**
	Splicing connectors	68.8	57.6	65.4	73.8	63.3	**89.7**	89.3 ± 0.2
	Avg. Logical	65.9	60.1	66.4	71.0	69.0	86.0	**88.9 ± 0.6**
Structural Anoma.	Breakfast box	50.7	60.2	68.4	74.7	74.8	80.9	**87.5 ± 0.1**
	Juice bottle	77.8	84.1	**99.3**	84.9	86.7	98.9	93.1 ± 0.3
	Pushpins	74.9	76.7	**90.3**	58.1	77.6	74.9	74.2 ± 17.4
	Screw bag	46.1	56.8	87.0	59.8	86.6	70.5	**92.2 ± 0.8**
	Splicing connectors	63.8	73.2	**96.8**	57.1	68.7	78.3	76.7 ± 0.2
	Avg. Structural	62.7	70.2	**88.3**	66.9	78.9	80.7	84.7 ± 3.4
Avg. Total		64.3	65.1	77.4	68.9	74.0	83.4	**86.8 ± 1.8**

one example, an anomaly may include a different number of objects than the numbers expected from a normal sample (while all the featured object types exist in the normal class (Fig.1)). Other types of logical anomalies in the dataset may include cases where distant parts of an image must correlate with one another. E.g., within the normal data, the color of one object may correlate with the length of another object. These correlations may break in an anomalous sample.

Metric. Following the standard in image-level anomaly detection we report image level ROC-AUC metric. We report this metric individually for each anomaly type, for each data class.

Baselines. We compare to baseline methods used by the paper which presented the MVTec-LOCO dataset Bergmann et al. (2022). Namely, we compare to *Variational Model (VM)* Steger (2001) - a handcrafted similarity measure designed for robustness to different conditions, *MNAD* Park et al. (2020) - an autoencoder with a memory module, *f-AnoGAN* Schlegl et al. (2017) - a generative model trained for the reconstruction of anomaly free images, *AE / VAE* Sakurada & Yairi (2014) - an autoencoder / variational autoencoder, *Student Teacher* (ST) Bergmann et al. (2020) - a student network aimed to give better reconstruction for the normal data, *SPADE* Cohen & Hoshen (2020) - a density estimation method using a pyramid of deep ResNet features, *PatchCore* (PCore) Roth et al. (2022) - a state-of-the-art method for structural anomalies, improving SPADE scoring function, *GCAD* Bergmann et al. (2022) - a reconstruction based method, based on both local and global deep ResNet features.

Results. We report our results on image-level detection of logical anomalies and structural anomalies in Tab.1. Interestingly, we find complementary strengths between our approach and GCAD, a reconstruction-based approach by Bergmann et al. (2022). Although GCAD performed better on specific classes (e.g., *pushpins*), our approach provides better results on average. Most notably, our approach provides non-trivial anomaly detection capabilities on the *screw bag* class, while baseline approaches are close to the random baseline. The rest of the compared approaches performed significantly worse on all logical anomaly classes, as they rely on the abnormality of single patches.

Our approach also provides an improvement in the detection of structural anomalies in some classes. This is somewhat surprising, as one may assume that detection-by-segmentation approaches would perform well in these cases. One possible reason for that is the high variability of the normal data in some of the classes (e.g., *breakfast box*, *screw bag*, Fig.1). This high variability may induce false positive detections for detection-by-segmentation approaches. Taken together, while different methods provide complementary strengths, on average, our method provides state-of-the-art results on logical anomaly detection, and on the dataset as a whole.

5.2 Time series anomalies detection results

Time series dataset. We compared on the five datasets used in NeurTraL-AD Qiu et al. (2021): *RacketSports (RS).* Accelerometer and gyroscope recording of players playing four different racket sports. Each sport is designated as a different class. *Epilepsy (EPSY).* Accelerometer recording of healthy actors simulating four different activity classes, one of them being an epileptic shock.

Table 2: Anomaly detection on the UEA datasets, average ROC-AUC (%) over all classes. See Tab.6 for the full table. σ presented in Tab. 7

	OCSVM	IF	RNN	ED	DSVDD	DAG	GOAD	DROCC	NeuTraL	Ours
EPSY	61.1	67.7	80.4	82.6	57.6	72.2	76.7	85.8	92.6	**98.1**
NAT	86.0	85.4	89.5	91.5	88.6	78.9	87.1	87.2	94.5	**96.1**
SAD	95.3	88.2	81.5	93.1	86.0	80.9	94.7	85.8	**98.9**	97.8
CT	97.4	94.3	96.3	79.0	95.7	89.8	97.7	95.3	99.3	**99.7**
RS	70.0	69.3	84.7	65.4	77.4	51.0	79.9	80.0	86.5	**92.3**
Avg.	82.0	81.0	86.5	82.3	81.1	74.6	87.2	86.8	94.4	**96.8**

Naval air training and operating procedures standardization (NAT). Positions of sensors mounted on different body parts of a person performing activities. There are six different activity classes in the dataset. *Character trajectories (CT).* Velocity trajectories of a pen on a WACOM tablet. There are 20 different characters in this dataset. *Spoken Arabic Digits (SAD).* MFCC features of ten Arabic digits spoken by 88 different speakers.

Metric. Following Qiu et al. (2021), we use the series-level ROC-AUC metric.

Baselines. We compare the results of several baseline methods reported by Qiu et al. (2021). The methods cover the following paradigms: *One-class classification*: One-class SVM (OC-SVM), and its deep versions DeepSVDD ("DSVDD") Ruff et al. (2018), and the recently published DROCC Goyal et al. (2020). *Tree-based detectors*: Isolation Forest (IF) Liu et al. (2008). *Density estimation*: LOF, a specialized version of nearest neighbor anomaly detection Breunig et al. (2000). DAGMM ("DAG") Zong et al. (2018): density estimation in an auto-encoder latent space *Auto-regressive methods* - RNN and LSTM-ED ("ED") - deep neural network-based version of auto-regressive prediction models Malhotra et al. (2016). *Transformation prediction* - GOAD Bergman & Hoshen (2020) and NeuTraL-AD Qiu et al. (2021) are based on transformation prediction, and are adaptations of RotNet-based approaches (such as GEOM Golan & El-Yaniv (2018)).

Results. Our results are presented in Tab. 2. We can observe that different baseline approaches are effective for different datasets. kNN-based LOF is highly effective for SAD which is a large dataset but achieves worse results for EPSY. Auto-regressive approaches achieve strong results on CT. Transformation-prediction approaches, GOAD and NeuTraL achieve the best performance of all the baselines. The learned transformations of NeuTraL achieved better results than the random transformations of GOAD.

Our method achieves the best overall results both on average and individually on all datasets apart from SAD (where it is comparable but a little lower than NeuTraL). Note that differently from NeuTraL, our method is far simpler, does not use deep neural networks and is very fast to train and evaluate. It also has fewer hyperparameters.

5.3 Implementation Details

We provide here the main implementation details for our image anomaly detection application. Further implementation details for the image application can be found in App.C.1. Implementation details for the time series experiments can be found in App.C.2.

ResNet levels. We use the representations from the third and fourth blocks of a *WideResNet50×2* (resulting in sets size 7×7 and 14×14 elements, respectively). We also use all the raw pixels in the image as an additional set (resized to 224×224 elements).

The total anomaly score is the average of the anomaly scores obtained for the set of 3rd ResNet block features, the set of 4th ResNet block features, and the set of raw pixels. The average anomaly score is weighted by the following factors $(1, 1, 0.1)$ respectively (see App.D for our robustness to the choice of weighting factor).

Multiple crops for image anomaly detection. Describing the entire image as a single set might sometime lose discriminative power when the anomalies are localized. To mitigate this issue, we can treat only a part of an image as our entire set. To do so, we crop the image to a factor of c, and compare the elements taken only from these crops. We compute an anomaly score for each crop

Table 3: Ablation for logical image AD. ROC-AUC (%) .

	Only 3	Only 4	No pixels	Only full	Ours
Breakfa.	93.9	95.6	95.7	**97.1**	96.5
Juice bo.	93.3	**97.4**	96.4	96.7	96.6
Pushpins	78.0	66.8	73.4	77.5	**83.4**
Screw b.	**79.5**	71.0	77.2	76.6	78.6
Splicing.	85.3	85.0	86.3	**91.3**	89.3
Average	86.0	83.2	85.8	87.8	**88.9**

Table 4: Ablation for logical image AD. Compared using no multiple crops and no raw-pixels level. ROC-AUC (%).

	Sim. Avg.	No Proj.	No Whit.	Ours
Breakfa.	88.4	91.2	94.2	**95.1**
Juice bo.	95.7	94.7	93.4	**95.2**
Pushpins	64.1	68.9	69.6	**73.4**
Screw b.	56.5	62.3	66.4	**70.2**
Splicing.	87.7	**89.2**	88.0	86.7
Average	78.5	81.3	82.3	**84.1**

factor and for each center location. We then average over the anomaly scores of the different crop center locations for the same crop factor c. Finally, for each ResNet level (described above), we average the anomaly scores over the different crop ratios c. We use crop ratios of $\{1.0, 0.7, 0.5, 0.33\}$. The different center locations are taken with a stride of 0.25 of the entire image.

Parameters. For the image experiments, we use histograms of $K = 5$ bins and $r = 1000$ projections. For the raw-pixels layer, we used a projection dimension of $r = 10$ and no whitening due to its low number of channels. To avoid high variance between runs, we averaged 16 different repetitions for the raw-pixel scoring. We use $k = 1$ for the kNN density estimation.

5.4 Ablations

Here, we present ablations for the image logical AD methods. For time series ablations, see appendix E.

Using individual ResNet levels. In Tab.3 we report the results of our method when different components of our multi-level ResNet ensemble are removed. We report the results using only the representation from the third or fourth ResNet block ("Only 3 / 4"). We also report the results of using both ResNet blocks, but without the raw-pixels level ("No Pixels"). While specific variants may outperform on specific classes, our combination outperforms on average.

No multiple crops ablation. We also report our results without the multiple crops ensemble (described in Sec.5.3). In this ablation we feed only the entire image for the set extraction stage ("Only full"). As expected, using multiple crops field is beneficial for classes where small components are relatively important.

Simple averaging. We compare to a simple averaging of the set features (Fig. 2), ablating our entire set-features approach. This yields a significantly worse performance (Tab.4 "Sim. Avg.").

No random projection. We also ablate our use of random projections as described in Sec.3.3. We replace the random histograms with similar histograms using the raw given features. The advantage of our approach can be seen in Tab.4 ("No Proj.").

No whitening. Finally, we also ablate our Gaussian model of the set features. Our method uses the collection of all the normal samples to calculate the covariance of the set features of the normal sample and modify our set distance measure as described in Sec.3.4. Also here our approach outperforms, highlighting the benefits of describing the normal data as a collection of sets (Tab.4 "No Whit.").

6 Discussion

Complementary strength of density estimation and reconstruction based approaches for logical anomaly detection. As our method and GCAD Bergmann et al. (2022), a reconstruction based approach, exhibit complementary strengths, it is a natural direction to try and use them together. A practical way to take advantage of both approaches would be ensembling. A better understanding of the reasons for each method's different performance across classes is likely to lead to the development of better approaches, combining the strengths of each method.

Is our set descriptor approach beneficial for detecting structural image anomalies? While our method slightly lags behind the top segmentation-by-detection approach on structural anomalies, it achieves the top performance on specific classes. We hypothesize this may be due to the high variation among the normal samples in these classes. In this case too, future research may allow the construction of better detectors, enjoying the combined strength of both approaches.

Incorporating deep features for time series data. We demonstrated that our method is able to outperform the state-of-the-art in time series anomaly detection without using deep neural networks. While this is an interesting and surprising result, we believe that deep features will be incorporated into similar approaches in the future. One direction for doing this is replacing the window projection features with a suitable deep representation, while keeping the averaging and Gaussian modeling steps unchanged.

Relation to previous methods and optimal transport. Our method is related to several previous methods. HBOS (Goldstein & Dengel, 2012) and LODA (Pevnỳ, 2016) also used similar projection features for anomaly detection. Yet, these methods perform histogram-based density estimation by ignoring the dependency across projections. As they can only be applied to a single element (time-window), they do not achieve competitive performance for time series AD. Rocket/mini-rocket (Dempster et al., 2020, 2021) also average projection features across windows but do not tackle anomaly detection nor do they apply to image data. Finally, there is a subtle connection to Radon transform (Kolouri et al., 2015) and sliced Wasserstein distance in L_1 (SWD) based methods (Bonneel et al., 2015) which also use similar projection and histogram features.

7 Limitations

Element-level anomaly detection. Our method focuses on sample-level time series and image-level anomaly detection. In some applications, a user may also want a segmentation of the most anomalous elements of each sample. We note that for logical anomalies, this is often not very well defined. E.g., when we have an image with 3 nuts as opposed to the normal 2, each of them may be considered anomalous. To provide element-level information, our method can be combined with current segmentation approaches by incorporating the knowledge of a global anomaly (e.g., removing false positive segmentation if an image is normal). However, directly applying our set features for anomaly segmentation is left for future research.

Pre-trained features. Similarly to the other top-performing approaches, our approach for image anomaly detection relies on pre-trained features. While the use of pre-trained features for anomaly detection in images is standard, it has failure modes. There are a handful of datasets where ImageNet pretraining is known to fail Yousef et al. (2023).

Class-specific performance. While our method outperforms on average, in some classes we do not perform as well compared to baseline approaches. A better understanding of the cases when our method fails would be beneficial for deploying it in practice.

Non-IID elements. Our method uses a Gaussian anomaly scoring function. By the central limit theorem, this is justifiable when the elements are IID (see appendix F). However, this is not precisely true for either of our settings as elements have a strong overlap. In general, we find the Mahalanobis scoring function is highly effective, and the combination with kNN relaxes the Gaussian assumption. A more careful analysis is left for future work.

8 Conclusion

We presented a method for detecting anomalies caused by unusual combinations of normal elements. We introduce set features dedicated to capturing such phenomena, and demonstrate their applicability for images and time series. Extensive experiments established the strong performance of our method.

9 Acknowledgement

Niv Cohen was funded by Israeli Science Foundation and the Hebrew University Data Science grants (CIDR). We thank Paul Bergmann for kindly sharing numerical results for many of the methods compared on the MVTec-LOCO dataset.

References

Bergman, L. and Hoshen, Y. Classification-based anomaly detection for general data. In *International Conference on Learning Representations*, 2020.

Bergmann, P., Fauser, M., Sattlegger, D., and Steger, C. Mvtec ad–a comprehensive real-world dataset for unsupervised anomaly detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9592–9600, 2019.

Bergmann, P., Fauser, M., Sattlegger, D., and Steger, C. Uninformed students: Student-teacher anomaly detection with discriminative latent embeddings. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4183–4192, 2020.

Bergmann, P., Batzner, K., Fauser, M., Sattlegger, D., and Steger, C. The mvtec anomaly detection dataset: a comprehensive real-world dataset for unsupervised anomaly detection. *International Journal of Computer Vision*, 129(4):1038–1059, 2021.

Bergmann, P., Batzner, K., Fauser, M., Sattlegger, D., and Steger, C. Beyond dents and scratches: Logical constraints in unsupervised anomaly detection and localization. *International Journal of Computer Vision*, 130(4):947–969, 2022.

Blázquez-García, A., Conde, A., Mori, U., and Lozano, J. A. A review on outlier/anomaly detection in time series data. *ACM Computing Surveys (CSUR)*, 54(3):1–33, 2021.

Bonneel, N., Rabin, J., Peyré, G., and Pfister, H. Sliced and radon wasserstein barycenters of measures. *Journal of Mathematical Imaging and Vision*, 51(1):22–45, 2015.

Bontemps, L., Cao, V. L., McDermott, J., and Le-Khac, N.-A. Collective anomaly detection based on long short-term memory recurrent neural networks. In *International conference on future data and security engineering*, pp. 141–152. Springer, 2016.

Breunig, M. M., Kriegel, H.-P., Ng, R. T., and Sander, J. Lof: identifying density-based local outliers. In *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, pp. 93–104, 2000.

Carrera, D., Manganini, F., Boracchi, G., and Lanzarone, E. Defect detection in sem images of nanofibrous materials. *IEEE Transactions on Industrial Informatics*, 13(2):551–561, 2016.

Cohen, M. J. and Avidan, S. Transformaly-two (feature spaces) are better than one. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4060–4069, 2022.

Cohen, N. and Hoshen, Y. Sub-image anomaly detection with deep pyramid correspondences. *arXiv preprint arXiv:2005.02357*, 2020.

Dasgupta, A., Kumar, R., and Sarlós, T. Fast locality-sensitive hashing. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 1073–1081, 2011.

Deecke, L., Ruff, L., Vandermeulen, R. A., and Bilen, H. Transfer-based semantic anomaly detection. In *International Conference on Machine Learning*, pp. 2546–2558. PMLR, 2021.

Defard, T., Setkov, A., Loesch, A., and Audigier, R. Padim: a patch distribution modeling framework for anomaly detection and localization. In *Pattern Recognition. ICPR International Workshops and Challenges: Virtual Event, January 10–15, 2021, Proceedings, Part IV*, pp. 475–489. Springer, 2021.

Dempster, A., Petitjean, F., and Webb, G. I. Rocket: exceptionally fast and accurate time series classification using random convolutional kernels. *Data Mining and Knowledge Discovery*, 34(5): 1454–1495, 2020.

Dempster, A., Schmidt, D. F., and Webb, G. I. Minirocket: A very fast (almost) deterministic transform for time series classification. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pp. 248–257, 2021.

Eskin, E., Arnold, A., Prerau, M., Portnoy, L., and Stolfo, S. A geometric framework for unsupervised anomaly detection. In *Applications of data mining in computer security*, pp. 77–101. Springer, 2002.

Glodek, M., Schels, M., and Schwenker, F. Ensemble gaussian mixture models for probability density estimation. *Computational Statistics*, 28(1):127–138, 2013.

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

Goldstein, M. and Dengel, A. Histogram-based outlier score (hbos): A fast unsupervised anomaly detection algorithm. *KI-2012: Poster and Demo Track*, 2012.

Goyal, S., Raghunathan, A., Jain, M., Simhadri, H. V., and Jain, P. Drocc: Deep robust one-class classification. In *International Conference on Machine Learning*, pp. 3711–3721. PMLR, 2020.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Hendrycks, D., Mazeika, M., Kadavath, S., and Song, D. Using self-supervised learning can improve model robustness and uncertainty. In *Advances in Neural Information Processing Systems*, pp. 15663–15674, 2019.

Horwitz, E. and Hoshen, Y. An empirical investigation of 3d anomaly detection and segmentation. *arXiv preprint arXiv:2203.05550*, 2022.

Jezek, S., Jonak, M., Burget, R., Dvorak, P., and Skotak, M. Deep learning-based defect detection of metal parts: evaluating current methods in complex conditions. In *2021 13th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT)*, pp. 66–71. IEEE, 2021.

Johnson, J., Douze, M., and Jégou, H. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3):535–547, 2019.

Kolouri, S., Park, S. R., and Rohde, G. K. The radon cumulative distribution transform and its application to image classification. *IEEE transactions on image processing*, 25(2):920–934, 2015.

Larsson, G., Maire, M., and Shakhnarovich, G. Learning representations for automatic colorization. In *ECCV*, 2016.

Latecki, L. J., Lazarevic, A., and Pokrajac, D. Outlier detection with kernel density functions. In *International Workshop on Machine Learning and Data Mining in Pattern Recognition*, pp. 61–75. Springer, 2007.

Li, C.-L., Sohn, K., Yoon, J., and Pfister, T. Cutpaste: Self-supervised learning for anomaly detection and localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9664–9674, 2021.

Liu, F. T., Ting, K. M., and Zhou, Z.-H. Isolation forest. In *2008 Eighth IEEE International Conference on Data Mining*, pp. 413–422. IEEE, 2008.

Malhotra, P., Ramakrishnan, A., Anand, G., Vig, L., Agarwal, P., and Shroff, G. Lstm-based encoder-decoder for multi-sensor anomaly detection. *arXiv preprint arXiv:1607.00148*, 2016.

Park, H., Noh, J., and Ham, B. Learning memory-guided normality for anomaly detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14372–14381, 2020.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Perera, P. and Patel, V. M. Learning deep features for one-class classification. *IEEE Transactions on Image Processing*, 28(11):5450–5463, 2019.

Pevnỳ, T. Loda: Lightweight on-line detector of anomalies. *Machine Learning*, 102(2):275–304, 2016.

Qiu, C., Pfrommer, T., Kloft, M., Mandt, S., and Rudolph, M. Neural transformation learning for deep anomaly detection beyond images. *ICML*, 2021.

Radon, J. 1.1 über die bestimmung von funktionen durch ihre integralwerte längs gewisser mannigfaltigkeiten. *Proceedings of the Royal Saxonian Academy of Sciences at Leipzig*, 1917.

Reiss, T. and Hoshen, Y. Mean-shifted contrastive loss for anomaly detection. *arXiv preprint arXiv:2106.03844*, 2021.

Reiss, T., Cohen, N., Bergman, L., and Hoshen, Y. Panda: Adapting pretrained features for anomaly detection and segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2806–2814, 2021.

Reiss, T., Cohen, N., Horwitz, E., Abutbul, R., and Hoshen, Y. Anomaly detection requires better representations. *arXiv preprint arXiv:2210.10773*, 2022.

Roth, K., Pemula, L., Zepeda, J., Schölkopf, B., Brox, T., and Gehler, P. Towards total recall in industrial anomaly detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14318–14328, 2022.

Rousseeuw, P. J. and Leroy, A. M. *Robust regression and outlier detection*, volume 589. John wiley & sons, 2005.

Ruff, L., Vandermeulen, R., Goernitz, N., Deecke, L., Siddiqui, S. A., Binder, A., Müller, E., and Kloft, M. Deep one-class classification. In *International conference on machine learning*, pp. 4393–4402, 2018.

Ruff, L., Vandermeulen, R. A., Görnitz, N., Binder, A., Müller, E., Müller, K.-R., and Kloft, M. Deep semi-supervised anomaly detection. In *International Conference on Learning Representations*, 2019.

Ruff, L., Kauffmann, J. R., Vandermeulen, R. A., Montavon, G., Samek, W., Kloft, M., Dietterich, T. G., and Müller, K.-R. A unifying review of deep and shallow anomaly detection. *Proceedings of the IEEE*, 2021.

Sakurada, M. and Yairi, T. Anomaly detection using autoencoders with nonlinear dimensionality reduction. In *Proceedings of the MLSDA 2014 2nd workshop on machine learning for sensory data analysis*, pp. 4–11, 2014.

Salehi, M., Sadjadi, N., Baselizadeh, S., Rohban, M. H., and Rabiee, H. R. Multiresolution knowledge distillation for anomaly detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 14902–14912, 2021.

Schlegl, T., Seeböck, P., Waldstein, S. M., Schmidt-Erfurth, U., and Langs, G. Unsupervised anomaly detection with generative adversarial networks to guide marker discovery. In *International conference on information processing in medical imaging*, pp. 146–157. Springer, 2017.

Schölkopf, B., Platt, J. C., et al. Support vector method for novelty detection. Citeseer.

Steger, C. Similarity measures for occlusion, clutter, and illumination invariant object recognition. In *Joint Pattern Recognition Symposium*, pp. 148–154. Springer, 2001.

Tack, J., Mo, S., Jeong, J., and Shin, J. Csi: Novelty detection via contrastive learning on distributionally shifted instances. *NeurIPS*, 2020.

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

Yousef, M., Ackermann, M., Kurup, U., and Bishop, T. No shifted augmentations (nsa): compact distributions for robust self-supervised anomaly detection. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 5511–5520, 2023.

Zaheer, M., Kottur, S., Ravanbakhsh, S., Poczos, B., Salakhutdinov, R. R., and Smola, A. J. Deep sets. *Advances in neural information processing systems*, 30, 2017.

Zong, B., Song, Q., Min, M. R., Cheng, W., Lumezanu, C., Cho, D., and Chen, H. Deep autoencoding gaussian mixture model for unsupervised anomaly detection. In *International Conference on Learning Representations*, 2018.

A Full Results Tables

The full table image logical anomaly detection experiments can be found in Tab.5. The full table for the time series anomaly detection experiments can be found in Tab.6.

B UEA Results with Standard Errors

We present an extended version of the UEA results including error bounds for our method and baselines that reported them. The difference between the methods is significantly larger than the standard error.

C Implementation Details

Histograms. In practice, we use the cumulative histograms as our set features for both data modalities (of Sec.3.3).

C.1 Image anomaly detection

Preprocessing. Before feeding each image sample to the pre-trained network we resize it to 224×224 and normalize it according to the standard ImageNet mean and variance.

Considering that classes in this dataset are provided in different aspect ratios, and that similar objects may look different when resized to a square, we found it beneficial to pad each image with empty pixels. The padded images have a $1:1$ aspect ratio, and resizing them would not change the aspect ratio of the featured objects.

Software. For the whitening of image features we use the *ShrunkCovariance* function from the *scikit-learn library* Pedregosa et al. (2011) with its default parameters. For kNN density estimation we use the *faiss* library Johnson et al. (2019).

Computational resources. The experiments were run on a single RTX2080-GT GPU.

C.2 Time Series anomaly detection

Padding. Prior to window extraction, the series x is first right and left zero-padded by $\frac{\tau}{2}$ to form padded series x'. The first window w_1 is defined as the first τ observations in padded series S', i.e. $w_1 = x'_1, x'_2..x'_\tau$. We further define windows at higher scales W^s, which include observations sampled with stride c. At scale c, the original series x is right and left zero-padded by $\frac{c \cdot \tau}{2}$ to form padded series S'^c.

UEA Experiments. We used each time series as an individual training sample. We chose a kernel size of 9, 100 projection, 20 quantiles, and a maximal number of levels of 10. The results varied only slightly within a reasonable range of the hyperparameters e.g. using 5, 10, 15 levels yielded an average ROCAUC of 97, 96.8, 96.8 across the five UEA datasets.

Spoken Arabic Digits processing We follow the processing of the dataset as done by Qiu et al. Qiu et al. (2021). In private communications the authors explained that only sequences of lengths between 20 and 50 time steps were selected. The other time series were dropped.

Computational resources. The experiments were run on a modest number of CPUs on a computing cluster. The baseline methods were run on a single RTX2080-GT GPU

D Logical Anomaly detection Robustness

We check the robustness of our results for the parameter λ - the weighting between the raw-pixels level anomaly score to the anomaly score derived from the ResNet features (Sec.5.3). As can be seen in Tab.8, our results are robust to the choice of λ.

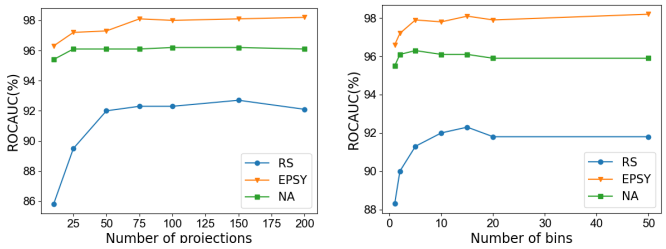

Figure 4: Ablation of accuracy vs. the number of projections (left) and the number of bins (right).

E Time Series Ablations

Number of projections. Using a high output dimension for projection matrix P increases the expressively but also increases the computation cost. We investigate the effect of the number of projections on the final accuracy of our method. The results are provided in Fig. 4. We can observe that although a small number of projections hurts performance, even a moderate number of projections is sufficient. We found 100 projections to be a good tradeoff between performance and runtime.

Number of bins. We compute the accuracy of our method as a function of the number of bins per projection. Our results (Fig. 4) show that beyond a very small number of bins - larger numbers are not critical. We found 20 bins to be sufficient in all our experiments.

Effect of Gaussian density estimation. Standard projection methods such as HBOS Goldstein & Dengel (2012) and LODA Pevnỳ (2016) do not use a multivariate density estimator but instead estimate the density of each dimension independently. We compare using a full and per-variable density estimation in Tab. 9. We can see that our approach achieves far better results, attesting to the importance of modeling the correlation between projections.

Comparing projection sampling methods. We compare three different projection selection procedures: (i) Gaussian: sampling the weights in P from a random Normal Gaussian distribution (ii) Using an identity projection matrix: $P = I$. (iii) PCA: selecting P from the eigenvectors of the matrix containing all (raw) features of all training windows. PCA selects the projections with maximum variation but is computationally expensive. The results are presented in Tab. 10. We find that the identity projection matrix under-performed the other approaches (as it provides no variable mixing). Surprisingly, we do not see a large difference between PCA and random projections.

F Using the Central Limit Theorem for Set anomaly detection

We model the features of each window f as a normal set as IID observations coming from a probability distribution function $p(f)$. The distribution function is *not* assumed to be Gaussian. Using a Gaussian density estimator trained on the features of elements observed in training is unlikely to be effective for element-level anomaly detection (due to the non-Gaussian $p(f)$).

An alternative formulation to the one presented in section 3, is that each feature f is multiplied by projection matrix P, and then each dimension is discretized and mapped to a one-hot vector. This formulation therefore maps the representation of each element to a sparse binary vector. The mean of the representations of elements in the set recovers the normalized histogram descriptor precisely (therefore this formulation is equivalent to the one in section 3). As the histogram is a mean of the set of elements, it has superior statistical properties. In particular, the Central Limit Theorem states that under some conditions the sample mean follows the Gaussian distribution regardless of the distribution of windows $p(f)$. While typically in anomaly detection only a single sample is presented at a time, the situation is different when treating samples as sets. Although the windows are often not IID, given a multitude of elements, an IID approximation may be approximately correct. This explains the high effectiveness of Gaussian density estimation in our formulation.

Table 5: Anomaly detection on the MVTec-LOCO dataset. ROC-AUC (%).

		VM	AE	VAE	f-AG	MNAD
Logical Anomalies	Breakfast box	70.3	58.0	47.3	69.4	59.9
	Juice bottle	59.7	67.9	61.3	82.4	70.5
	Pushpins	42.5	62.0	54.3	59.1	51.7
	Screw bag	45.3	46.8	47.0	49.7	60.8
	Splicing connectors	64.9	56.2	59.4	68.8	57.6
	Avg. Logical	56.5	58.2	53.8	65.9	60.1
Structural Anom.	Breakfast box	70.1	47.7	38.3	50.7	60.2
	Juice bottle	69.4	62.6	57.3	77.8	84.1
	Pushpins	65.8	66.4	75.1	74.9	76.7
	Screw bag	37.7	41.5	49.0	46.1	56.8
	Splicing connectors	51.6	64.8	54.6	63.8	73.2
	Avg. Structural	58.9	56.6	54.8	62.7	70.2
Avg. Total		57.7	57.4	54.3	64.3	65.1

		ST	SPADE	PCore	GCAD	SINBAD
Logical Anomalies	Breakfast box	68.9	81.8	77.7	87.0	**96.5 ± 0.1**
	Juice bottle	82.9	91.9	83.7	**100.0**	96.6 ± 0.1
	Pushpins	59.5	60.5	62.2	**97.5**	83.4 ± 3.0
	Screw bag	55.5	46.8	55.3	56.0	**78.6 ± 0.1**
	Splicing connectors	65.4	73.8	63.3	**89.7**	89.3 ± 0.2
	Avg. Logical	66.4	71.0	69.0	86.0	**88.9 ± 0.6**
Structural Anom.	Breakfast box	68.4	74.7	74.8	80.9	**87.5 ± 0.1**
	Juice bottle	**99.3**	84.9	86.7	98.9	93.1 ± 0.3
	Pushpins	**90.3**	58.1	77.6	74.9	74.2 ± 17.4
	Screw bag	87.0	59.8	86.6	70.5	**92.2 ± 0.8**
	Splicing connectors	**96.8**	57.1	68.7	78.3	76.7 ± 0.2
	Avg. Structural	**88.3**	66.9	78.9	80.7	84.7 ± 3.4
Avg. Total		77.4	68.9	74.0	83.4	**86.8 ± 1.8**

Table 6: UEA datasets, average ROC-AUC (%) over all classes. (σ presented in Tab. 7)

	OCSVM	IF	LOF	RNN	ED	
EPSY	61.1	67.7	56.1	80.4	82.6	
NAT	86.0	85.4	89.2	89.5	91.5	
SAD	95.3	88.2	98.3	81.5	93.1	
CT	97.4	94.3	97.8	96.3	79.0	
RS	70.0	69.3	57.4	84.7	65.4	
Avg.	82.0	81.0	79.8	86.5	82.3	

	DSVDD	DAGMM	GOAD	DROCC	NeuTraL	Ours
EPSY	57.6	72.2	76.7	85.8	92.6	**98.1**
NAT	88.6	78.9	87.1	87.2	94.5	**96.1**
SAD	86.0	80.9	94.7	85.8	**98.9**	97.8
CT	95.7	89.8	97.7	95.3	99.3	**99.7**
RS	77.4	51.0	79.9	80.0	86.5	**92.3**
Avg.	81.1	74.6	87.2	86.8	94.4	**96.8**

Table 7: UEA datasets, average ROC-AUC (%) over all classes including error bounds

	OCSVM	IF	LOF	RNN	LSTM-ED	
EPSY	61.1	67.7	56.1	80.4 ± 1.8	82.6 ± 1.7	
NAT	86	85.4	89.2	89.5 ± 0.4	91.5 ± 0.3	
SAD	95.3	88.2	98.3	81.5 ± 0.4	93.1 ± 0.5	
CT	97.4	94.3	97.8	96.3 ± 0.2	79.0 ± 1.1	
RS	70	69.3	57.4	84.7 ± 0.7	65.4 ± 2.1	
Avg.	82.0	81.0	79.8	86.5	82.3	
	DeepSVDD	DAGMM	GOAD	DROCC	NeuTraL	Ours
EPSY	57.6 ± 0.7	72.2 ± 1.6	76.7 ± 0.4	85.8 ± 2.1	92.6 ± 1.7	**98.1** ± 0.3
NAT	88.6 ± 0.8	78.9 ± 3.2	87.1 ± 1.1	87.2 ± 1.4	94.5 ± 0.8	**96.1** ± 0.1
SAD	86.0 ± 0.1	80.9 ± 1.2	94.7 ± 0.1	85.8 ± 0.8	**98.9** ± 0.1	97.8 ± 0.1
CT	95.7 ± 0.5	89.8 ± 0.7	97.7 ± 0.1	95.3 ± 0.3	99.3 ± 0.1	**99.7** ± 0.1
RS	77.4 ± 0.7	51.0 ± 4.2	79.9 ± 0.6	80.0 ± 1.0	86.5 ± 0.6	**92.3** ± 0.3
Avg.	81.1	74.6	87.2	86.8	94.4	**96.8**

Table 8: Robustness to the choice of λ. Average Results on the Logical Anomalies classes. Average ROC-AUC (%).

λ	0.2	0.1 (Ours)	0.05	0.02
	88.8	**88.9**	88.7	88.5

Table 9: An ablation of projection sampling methods. ROC-AUC (%).

	EPSY	RS	NA	CT	SAD
No whitening	62.1	70.9	93.6	98.5	78.8
Whitening	**98.1**	**92.3**	**96.1**	**99.7**	**97.8**

Table 10: An ablation of projection sampling methods. ROC-AUC (%).

	EPSY	RS	NA	CT	SAD
Id.	97.1	90.2	91.8	98.2	78.3
PCA	**98.2**	91.6	95.8	**99.7**	96.7
Rand	98.1	**92.3**	**96.1**	**99.7**	**97.8**