DSV: An Alignment Validation Loss for Self-supervised Outlier Model Selection

Jaemin Yoo, Yue Zhao, Lingxiao Zhao, and Leman Akoglu(✉)

Carnegie Mellon University, Pittsburgh, USA
{jaeminyoo,zhaoy,lingxiao}@cmu.edu, lakoglu@andrew.cmu.edu

Abstract. Self-supervised learning (SSL) has proven effective in solving various problems by generating internal supervisory signals. Unsupervised anomaly detection, which faces the high cost of obtaining true labels, is an area that can greatly benefit from SSL. However, recent literature suggests that tuning the hyperparameters (HP) of data augmentation functions is crucial to the success of SSL-based anomaly detection (SSAD), yet a systematic method for doing so remains unknown. In this work, we propose DSV (Discordance and Separability Validation), an unsupervised validation loss to select high-performing detection models with effective augmentation HPs. DSV captures the alignment between an augmentation function and the anomaly-generating mechanism with surrogate losses, which approximate the discordance and separability of test data, respectively. As a result, the evaluation via DSV leads to selecting an effective SSAD model exhibiting better alignment, which results in high detection accuracy. We theoretically derive the degree of approximation conducted by the surrogate losses and empirically show that DSV outperforms a wide range of baselines on 21 real-world tasks.

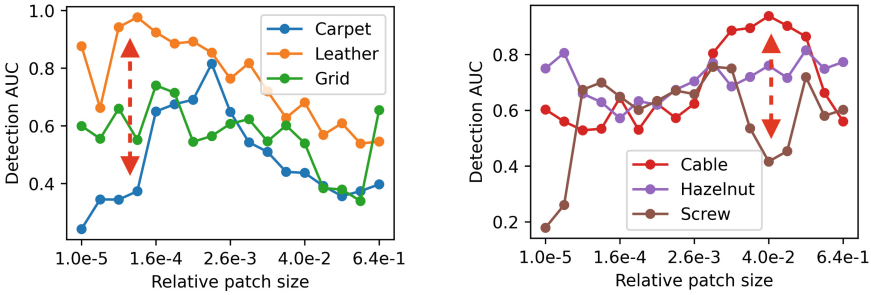
Keywords: Anomaly detection · Self-supervised learning · Unsupervised model selection · Data augmentation

1 Introduction

Through the use of carefully annotated data, machine learning (ML) has demonstrated success in various applications. Nonetheless, the high cost of acquiring high-quality labeled data poses a huge challenge. A recent alternative, known as self-supervised learning (SSL), has emerged as a promising solution. Intuitively, SSL generates a form of internal supervisory signal from the data to solve a task, thereby transforming an unsupervised task into a supervised problem by producing (pseudo-)labeled examples. It has achieved remarkable progress in advancing natural language processing [1,6] and computer vision tasks [4,12].

SSL can be particularly advantageous when dealing with unsupervised problems such as anomaly detection (AD). The process of labeling for such problems, such as correctly identifying fraudulent transactions, can be challenging and

D. Koutra et al. (Eds.): ECML PKDD 2023, LNAI 14169, pp. 254–269, 2023.
https://doi.org/10.1007/978-3-031-43412-9_15



(a) $f_{\text{aug}} = \text{CutOut}$ (b) $f_{\text{aug}} = \text{CutPaste}$

Fig. 1. The performance of self-supervised anomaly detectors on the MVTec AD dataset with different hyperparameters of augmentation f_{aug}. Each line is drawn from one of the 15 tasks that MVTec AD contains. The AUC changes from 0.242 to 0.815 based on the choice of hyperparameters (in Carpet), where the optimum depends on the type of f_{aug} and true anomalies.

expensive. As a result, a group of SSL-based AD (SSAD) approaches [2,7,13] have been proposed recently, where the core idea is to inject self-generated pseudo anomalies into the training data to improve the separability between inliers and pseudo anomalies. To create such pseudo anomalies, one may transform inliers via data augmentation function(s) such as rotate, blur, mask, or CutPaste [13], which are designed to create a systematic change without discarding important original properties such as texture or color depending on the dataset.

Despite the surge of SSAD methods, how to set the hyperparameters (HPs), e.g., rotation degrees, remain underexplored, which can significantly affect their performance [25]. In the supervised ML community, these augmentation HPs are systematically integrated into the model selection problem to be chosen with a hold-out/validation set [16,29]. However, choosing the augmentation HPs has been arbitrary and/or "cherry-picked" in SSAD [2,7] due to the evaluation challenges. Recent literature shows that the arbitrary choice of SSAD augmentation has implications [25]. Firstly, due to the no-free-lunch theorem [23], different augmentation techniques perform better on different detection tasks, and arbitrary selection is thus insufficient. Secondly, in some cases, the arbitrary selection of augmentation HPs can lead to a biased error distribution [24]. Thus, augmentation HPs in SSAD should be chosen carefully and systematically.

Figure 1 shows how the performance of SSAD methods changes by the choice of augmentation HPs. The CutOut [5] and CutPaste [13] augmentations are used for MVTec AD [3], which is a real-world dataset for anomaly detection. In Carpet of Fig. 1a, for example, the detection AUC changes from 0.242 to 0.815 with the choice of HPs. The expected accuracy without prior knowledge is severely worse than its optimum, highlighting the importance of a proper HP choice, which is not even the same for different augmentation functions and tasks.

One solution is to select augmentation HPs in SSAD via unsupervised outlier model selection (UOMS) [26,27], which aims to choose a good AD model and its HPs for a new dataset without using any labels. Given an underlying AD model, we may pair it with different augmentation HPs to construct candidate models to find the best performing one. Existing UOMS approaches can be briefly split into two groups. The first group solely depends on the model's output or input data [15], while it cannot capitalize on the nature of SSAD. The second group uses learning-based approaches to select a model using the performances on (similar) historical datasets, while this prior information may be inaccessible.

In this work, we propose DSV (Discordance and Separability Validation), an unsupervised objective function that enables the search for optimal augmentation HPs without requiring true labels. The main idea of DSV is to decompose the *alignment* between data augmentation and true anomalies, which cannot be computed without labels but plays an essential role in estimating the detection performance, into *discordance* and *separability*. Since each of them reflects only a part of the original alignment, the decomposition allows us to devise surrogate losses which effectively approximate the alignment in combination.

We summarize our key contributions below:

- **Unsupervised validation loss for SSAD:** We propose DSV, an unsupervised validation loss for the search of best augmentation HPs in SSAD. The minimization of DSV leads to a high-performing AD model, which exhibits better alignment between augmentation and true anomalies.
- **Theoretical analysis:** We theoretically show that DSV is an effective approximation of the alignment between data augmentation and true anomalies, and its minimization leads to well-aligned augmentation HPs.
- **Extensive experiments:** We conduct extensive experiments on 21 different real-world tasks. DSV surpasses 8 baseline approaches, showing up to 12.2% higher average AUC than the simple average. We also perform diverse types of ablation and case studies to better understand the success of DSV.

Reproducibility. All of our implementation and datasets are publicly available at https://github.com/jaeminyoo/DSV.

2 Problem Definition and Related Works

2.1 Problem Definition

Let $f_{\text{aug}} : \mathbb{R}^m \to \mathbb{R}^m$ be a data augmentation function on m-dimensional data, such as the rotation of an image, which plays an important role in self-supervised anomaly detection (SSAD). Then, we aim to solve the unsupervised outlier model selection (UOMS) problem, focusing on the hyperparameters (HP) of f_{aug}, based on observations that choosing good HPs of f_{aug} is as important as selecting the detector model or f_{aug} itself. We formally define the problem as Problem 1.

Problem 1. Let $\mathcal{D}_{\text{trn}}$ be a set of normal data, and $\mathcal{D}_{\text{test}}$ be an unlabeled test set containing both normal data and anomalies. Given $\mathcal{D}_{\text{trn}}$, $\mathcal{D}_{\text{test}}$, and a set $\{\phi_i\}_i$ of

detector models, each of which is trained on $\mathcal{D}_{\text{trn}}$ with an augmentation function f_{aug} of different hyperparameters, our goal is to find the model ϕ^* that produces the highest detection accuracy on $\mathcal{D}_{\text{test}}$, without having true labels.

We also assume that every detector model $\phi = \phi_{\text{enc}} \circ \phi_{\text{dec}}$ which we consider for UOMS consists of an encoder $\phi_{\text{enc}} \in \mathbb{R}^m \to \mathbb{R}^l$ and a decoder $\phi_{\text{dec}} \in \mathbb{R}^l \to \mathbb{R}$, which is typical of most AD models based on deep neural networks.

2.2 Self-supervised Anomaly Detection (SSAD)

With the recent advance in self-supervised learning, SSAD has been widely studied as a promising alternative to unsupervised AD models. The main idea is to create pseudo-anomalies and inject them into a training set, which contains only normal data, to utilize supervised training schemes. For example, a popular way is to learn a binary classifier that divides normal and augmented data [13] or an n-way classifier that predicts the type of augmentation used [2,7]. Many SSAD methods have shown a great performance on real-world tasks [17,19,20,22].

However, most existing works on SSAD are based on arbitrary and/or cherry-picked choices of an augmentation function and its HPs. This is because AD does not contain a labeled validation set for a systematic HP search unlike in typical supervised learning. A recent work [25] pointed out such a limitation of existing works and showed that augmentation HPs, as well as the augmentation function itself, work as important hyperparameters that largely affect the performance on each task. Thus, a systematic approach for unsupervised HP search is essential to design generalizable and reproducible approaches for SSAD.

2.3 Unsupervised Outlier Model Selection (UOMS)

UOMS aims to select an effective model without using any labels. Clearly, choosing the augmentation hyperparameters (HPs) of an AD algorithm in SSAD can be considered a UOMS problem. In this case, a candidate model is defined as a pair of the underlying AD algorithm and augmentation HPs, and the goal is to choose the one that would achieve high detection rate on test data.

Existing UOMS approaches can be categorized into two groups. The first group uses internal performance measures (IPMs) that are based solely on the model's output and/or input data [15]. We adopt three top-performing IPMs reported in [15] as baselines (see §4.1). The second group consists of meta-learning-based approaches [26,27]. In short, they facilitate model selection for a new unsupervised task by leveraging knowledge from similar historical tasks/datasets. It is important to note that in this work we do not assume access to historical training data. Thus, learning-based UOMS approaches do not apply here.

3 Proposed Method

We introduce DSV (Discordance and Separability Validation), our unsupervised validation loss for the search of augmentation HPs in SSAD. The minimization of DSV leads to better alignment between data augmentation and true anomalies, which in turn results in higher accuracy on anomaly detection.

3.1 Definitions and Assumptions

We first introduce definitions and assumptions on which DSV is based. We start by defining set distance and projection functions. Note that by Definition 1, the set distance d satisfies the triangle inequality between three different sets.

Definition 1. *We define a set distance d as the average of all pairwise distances:* $d(\mathcal{A},\mathcal{B}) = \frac{1}{|\mathcal{A}||\mathcal{B}|}\sum_{\mathbf{a}\in\mathcal{A}}\sum_{\mathbf{b}\in\mathcal{B}}\|\mathbf{a}-\mathbf{b}\|$. *We also represent the vector distance as d for the brevity of notations:* $d(\mathbf{a},\mathbf{b}) := d(\{\mathbf{a}\},\{\mathbf{b}\})$.

Definition 2. *We define a projected norm as* $\mathrm{proj}(\mathbf{a},\mathbf{b},\mathbf{c}) = \frac{(\mathbf{c}-\mathbf{a})^\top(\mathbf{b}-\mathbf{a})}{\|(\mathbf{b}-\mathbf{a})\|}$. *The meaning of* proj *is the norm of $\mathbf{c}-\mathbf{a}$ projected onto the direction of $\mathbf{b}-\mathbf{a}$, using $\mathbf{a}$ as the anchor point. Note that* $\mathrm{proj}(\mathbf{a},\mathbf{b},\mathbf{c}) \le \|\mathbf{c}-\mathbf{a}\|$.

Then, we introduce an assumption on data embeddings. Recall that our detector $\phi = \phi_{\mathrm{enc}} \circ \phi_{\mathrm{dec}}$ contains an encoder function $\phi_{\mathrm{enc}} \in \mathbb{R}^m \to \mathbb{R}^l$. Let $\mathcal{Z}_{\mathrm{trn}}$ and $\mathcal{Z}_{\mathrm{test}}$ be sets of embeddings for training and test samples, respectively, such that $\mathcal{Z}_{\mathrm{trn}} = \{\phi_{\mathrm{enc}}(\mathbf{x}) \mid \mathbf{x} \in \mathcal{D}_{\mathrm{trn}}\}$ and $\mathcal{Z}_{\mathrm{test}} = \{\phi_{\mathrm{enc}}(\mathbf{x}) \mid \mathbf{x} \in \mathcal{D}_{\mathrm{test}}\}$. Let $\mathcal{Z}_{\mathrm{test}}^{(n)}$ and $\mathcal{Z}_{\mathrm{test}}^{(a)}$ be the normal and anomalous data in $\mathcal{Z}_{\mathrm{test}}$, respectively. We also define $\mathcal{Z}_{\mathrm{aug}} = \{\phi_{\mathrm{enc}}(f_{\mathrm{aug}}(\mathbf{x})) \mid \mathbf{x} \in \mathcal{D}_{\mathrm{trn}}\}$ as a set of augmented embeddings.

Assumption 1. *By convention, we assume that training normal and test normal data are generated from the same underlying distribution. Let $d(\mathcal{Z}_{\mathrm{trn}},\mathcal{Z}_{\mathrm{trn}}) = \sigma$. Then, $d(\mathcal{Z}_{\mathrm{test}}^{(n)},\mathcal{Z}_{\mathrm{test}}^{(n)}) = \sigma$ and $d(\mathcal{Z}_{\mathrm{trn}},\mathcal{Z}_{\mathrm{test}}^{(n)}) = \sigma+\epsilon$, where $\epsilon < \sigma$.*

3.2 Main Ideas: Discordance and Separability

Let $f_{\mathrm{gen}} \in \mathbb{R}^m \to \mathbb{R}^m$ be the underlying (unknown) anomaly-generating function in $\mathcal{D}_{\mathrm{test}}$, which transforms a normal data into an anomaly. We aim to find f_{aug} that maximizes the functional similarity between f_{aug} and f_{gen}, which we refer to *alignment* in this work. There are various ways to measure the alignment, but we focus on the embedding space, as it allows us to avoid the high dimensionality of real-world data. We informally define the extent of alignment as follows.

Proposition 1. *Data augmentation function f_{aug} is aligned with the anomaly-generating function f_{gen} if $\mathcal{L}_{\mathrm{ali}} = d(\mathcal{Z}_{\mathrm{aug}},\mathcal{Z}_{\mathrm{test}}^{(a)})$ is small.*

The problem is $\mathcal{L}_{\mathrm{ali}}$ cannot be computed without test labels. To extract $\mathcal{Z}_{\mathrm{test}}^{(a)}$ from $\mathcal{Z}_{\mathrm{test}}$ is as difficult as solving the anomaly detection problem itself. Then, *how can we approximate $\mathcal{L}_{\mathrm{ali}}$ without test labels?* We propose to decompose the alignment geometrically into *discordance* h_d and *separability* h_s as shown in Fig. 2. For an intuitive illustration, we assume that only one data exists in each set, e.g., $\mathcal{Z}_{\mathrm{trn}} = \{\mathbf{z}_{\mathrm{trn}}\}$. Then, the simplified definitions of h_d and h_s are given as

$$h_d(\mathbf{z}_{\mathrm{trn}},\mathbf{z}_{\mathrm{aug}},\mathbf{z}_{\mathrm{test}}^{(a)}) = \frac{d(\mathbf{z}_{\mathrm{trn}},\mathbf{z}_{\mathrm{test}}^{(a)}) + d(\mathbf{z}_{\mathrm{aug}},\mathbf{z}_{\mathrm{test}}^{(a)})}{d(\mathbf{z}_{\mathrm{trn}},\mathbf{z}_{\mathrm{aug}})} - 1 \tag{1}$$

$$h_s(\mathbf{z}_{\mathrm{trn}},\mathbf{z}_{\mathrm{aug}},\mathbf{z}_{\mathrm{test}}^{(a)}) = \frac{\mathrm{proj}(\mathbf{z}_{\mathrm{trn}},\mathbf{z}_{\mathrm{aug}},\mathbf{z}_{\mathrm{test}}^{(a)})}{d(\mathbf{z}_{\mathrm{trn}},\mathbf{z}_{\mathrm{aug}})}\,. \tag{2}$$

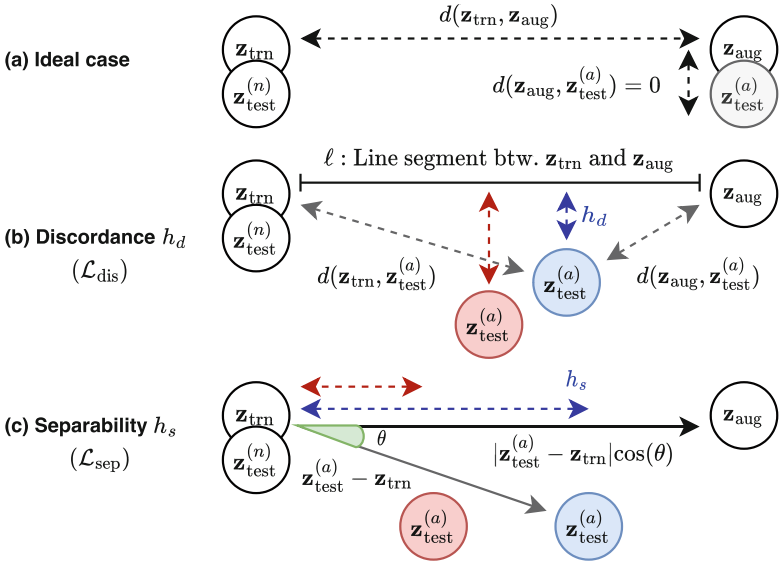


Fig. 2. Simplified illustrations of *discordance* and *separability*. We assume that all sets are of size one, e.g., $\mathcal{Z}_{\text{trn}} = \{\mathbf{z}_{\text{trn}}\}$. Blue is better than red in (b) and (c). To minimize $\mathcal{L}_{\text{ali}} = d(\mathbf{z}_{\text{aug}}, \mathbf{z}^{(a)}_{\text{test}})$ as in (a), we propose the (b) discordance h_d, which is the distance between $\mathbf{z}^{(a)}_{\text{test}}$ and the line segment ℓ, and the (c) separability h_s, which is the distance between $\mathbf{z}_{\text{trn}}$ and $\mathbf{z}^{(a)}_{\text{test}}$ projected onto ℓ. (Color figure online)

In combination, h_d and h_s allow us to minimize $\mathcal{L}_{\text{ali}} = d(\mathbf{z}_{\text{aug}}, \mathbf{z}^{(a)}_{\text{test}})$ without actually computing it. Let $\ell = \mathbf{z}_{\text{trn}} + t(\mathbf{z}_{\text{aug}} - \mathbf{z}_{\text{trn}})$ be a line segment between $\mathbf{z}_{\text{trn}}$ and $\mathbf{z}_{\text{aug}}$, where t ranges over $[0, 1]$. Then, h_d represents a distance between $\mathbf{z}^{(a)}_{\text{test}}$ and ℓ, which is minimized when $\mathbf{z}^{(a)}_{\text{test}}$ is exactly on ℓ. On the other hand, h_s means the distance between $\mathbf{z}^{(a)}_{\text{test}}$ and $\mathbf{z}_{\text{trn}}$ when $\mathbf{z}^{(a)}_{\text{test}}$ is projected onto ℓ. Thus, $\mathcal{L}_{\text{ali}}$ is minimized as zero if $h_d = 0$ and $h_s = 1$.

A difference between h_d and h_s is that h_d becomes a more accurate approximation of $\mathcal{L}_{\text{ali}}$ if $\mathbf{z}^{(a)}_{\text{test}}$ is far from both $\mathbf{z}_{\text{trn}}$ and $\mathbf{z}_{\text{aug}}$. Thus, we consider h_d as a coarse-grained measure, while we bound the range of h_s into $[0, 1]$ and consider it as a fine-grained measure to address the incapability of h_d to locate $\mathbf{z}^{(a)}_{\text{test}}$ on ℓ. Then, h_d is lower the better (alignment), while h_s is higher the better.

The exact definitions of h_d and h_s are direct generalization of Eq. (1) and (2) from vectors to sets. The idea is to compute the average of all possible distances by replacing the vector distance with the set distance in Definition 1:

$$h_d(\mathcal{Z}_{\text{trn}}, \mathcal{Z}_{\text{aug}}, \mathcal{Z}^{(a)}_{\text{test}}) = \frac{d(\mathcal{Z}_{\text{trn}}, \mathcal{Z}^{(a)}_{\text{test}}) + d(\mathcal{Z}_{\text{aug}}, \mathcal{Z}^{(a)}_{\text{test}})}{d(\mathcal{Z}_{\text{trn}}, \mathcal{Z}_{\text{aug}})} - 1 \tag{3}$$

$$h_s(\mathcal{Z}_{\text{trn}}, \mathcal{Z}_{\text{aug}}, \mathcal{Z}^{(a)}_{\text{test}}) = \frac{\sum_{\mathbf{z}_{\text{trn}}, \mathbf{z}_{\text{aug}}, \mathbf{z}^{(a)}_{\text{test}} \in \mathcal{Z}_{\text{trn}}, \mathcal{Z}_{\text{aug}}, \mathcal{Z}^{(a)}_{\text{test}}} \text{proj}(\mathbf{z}_{\text{trn}}, \mathbf{z}_{\text{aug}}, \mathbf{z}^{(a)}_{\text{test}})}{d(\mathcal{Z}_{\text{trn}}, \mathcal{Z}_{\text{aug}})|\mathcal{Z}_{\text{trn}}||\mathcal{Z}_{\text{aug}}||\mathcal{Z}^{(a)}_{\text{test}}|}. \tag{4}$$

Surrogate Losses. Based on our decomposition of the alignment, we propose surrogate losses $\mathcal{L}_{\text{dis}}$ and $\mathcal{L}_{\text{sep}}$ to approximate h_d and h_s, respectively, which have the term $\mathcal{Z}_{\text{test}}^{(a)}$ (unknown at test time) in their definitions. Our final validation loss $\mathcal{L}_{\text{DSV}}$ is given as

$$\mathcal{L}_{\text{DSV}}(\mathcal{Z}_{\text{trn}}, \mathcal{Z}_{\text{aug}}, \mathcal{Z}_{\text{test}}) = \mathcal{L}_{\text{dis}}(\cdot) - \frac{\max(\mathcal{L}_{\text{sep}}(\cdot), 1/2)}{\mathcal{L}_{\text{dis}}(\cdot)}, \tag{5}$$

where $\mathcal{Z}_{\text{trn}}$, $\mathcal{Z}_{\text{aug}}$, and $\mathcal{Z}_{\text{test}}$ are inputs also to the right-hand side terms. The minus sign is used since higher $\mathcal{L}_{\text{sep}}$ means better alignment until it reaches the optimum, which is $1/2$ in $\mathcal{L}_{\text{sep}}$, while it is 1 for h_s. We divide $\mathcal{L}_{\text{sep}}$ by $\mathcal{L}_{\text{dis}}$, since we want $\mathcal{L}_{\text{sep}}$ to have an effect especially when $\mathcal{L}_{\text{sep}}$ is small. Then, we use $\mathcal{L}_{\text{DSV}}$ to perform unsupervised model selection by choosing the hyperparameters of f_{aug} that yields the smallest $\mathcal{L}_{\text{DSV}}$, which indicates the model with best alignment.

3.3 Discordance Surrogate Loss

We now describe how our surrogate losses $\mathcal{L}_{\text{dis}}$ and $\mathcal{L}_{\text{sep}}$ effectively approximate the discordance h_d and separability h_s, respectively. $\mathcal{L}_{\text{dis}}$ is defined as

$$\mathcal{L}_{\text{dis}}(\mathcal{Z}_{\text{trn}}, \mathcal{Z}_{\text{aug}}, \mathcal{Z}_{\text{test}}) = \frac{d(\mathcal{Z}_{\text{trn}} \cup \mathcal{Z}_{\text{aug}}, \mathcal{Z}_{\text{test}})}{d(\mathcal{Z}_{\text{trn}}, \mathcal{Z}_{\text{aug}})}. \tag{6}$$

The idea is that $d(\mathcal{Z}_{\text{trn}} \cup \mathcal{Z}_{\text{aug}}, \mathcal{Z}_{\text{test}})$ can approximate h_d based on the triangle inequality. To show the exact relation between $\mathcal{L}_{\text{dis}}$ and h_d, we first derive the lower and upper bounds of $\mathcal{L}_{\text{dis}}$ with respect to h_d in Lemma 1. Then, we show in Corollary 1 that $\mathcal{L}_{\text{dis}}$ is represented as a linear function of h_d if some constraints are met, which makes $\mathcal{L}_{\text{dis}}$ an effective approximation of h_d.

Lemma 1. *If $|\mathcal{Z}_{\text{trn}}| = |\mathcal{Z}_{\text{aug}}|$, then the lower and upper bounds of $\mathcal{L}_{\text{dis}}$ are given as functions of h_d and $d(\mathcal{Z}_{\text{trn}}, \mathcal{Z}_{\text{aug}})$:*

$$c_2 h_d + c_2 + c_3 \leq \mathcal{L}_{\text{dis}}(\cdot) \leq c_2 h_d + c_2 + c_3 + \frac{(c_1 + c_3)(\sigma + \epsilon)}{d(\mathcal{Z}_{\text{trn}}, \mathcal{Z}_{\text{aug}})},$$

where $c_i = \hat{c}_i / \sum_{k=1}^{4} \hat{c}_k$ are data size-based constants such that $\hat{c}_1 = |\mathcal{Z}_{\text{trn}}| \cdot |\mathcal{Z}_{\text{test}}^{(n)}|$, $\hat{c}_2 = |\mathcal{Z}_{\text{trn}}| \cdot |\mathcal{Z}_{\text{test}}^{(a)}|$, $\hat{c}_3 = |\mathcal{Z}_{\text{aug}}| \cdot |\mathcal{Z}_{\text{test}}^{(n)}|$, and $\hat{c}_4 = |\mathcal{Z}_{\text{aug}}| \cdot |\mathcal{Z}_{\text{test}}^{(a)}|$.

Proof. The proof is in Appendix A.1. □

Corollary 1. *If $|\mathcal{Z}_{\text{trn}}| = |\mathcal{Z}_{\text{aug}}|$, $\sigma \ll d(\mathcal{Z}_{\text{trn}}, \mathcal{Z}_{\text{aug}})$, and $\epsilon \ll d(\mathcal{Z}_{\text{trn}}, \mathcal{Z}_{\text{aug}})$, then $\mathcal{L}_{\text{dis}}$ is a linear function of h_d: $\mathcal{L}_{\text{dis}}(\mathcal{Z}_{\text{trn}}, \mathcal{Z}_{\text{aug}}, \mathcal{Z}_{\text{test}}) \approx c_2 h_d + c_2 + c_3$.*

3.4 Separability Surrogate Loss

The separability surrogate loss $\mathcal{L}_{\text{sep}}$ for approximating h_s is defined as follows:

$$\mathcal{L}_{\text{sep}}(\cdot) = \frac{\text{std}(\{\text{proj}(\mu_{\text{trn}}, \mathbf{z}_{\text{aug}}, \mathbf{z}_{\text{test}}) \mid \mathbf{z}_{\text{aug}}, \mathbf{z}_{\text{test}} \in \mathcal{Z}_{\text{aug}}, \mathcal{Z}_{\text{test}}\})}{d(\mathcal{Z}_{\text{trn}}, \mathcal{Z}_{\text{aug}})}, \tag{7}$$

Table 1. Average AUC (top) and rank (bottom) across 21 different tasks in the two datasets. The best is in bold, and the second best is underlined. Our DSV achieves the best in six, and the second-best in two out of the 8 cases.

f_{aug}	Avg.	Rand.	Base	MMD	STD	MC	SEL	HITS	**DSV**
CutOut	0.739	<u>0.776</u>	0.741	0.735	0.739	0.749	0.727	0.757	**0.813**
CutAvg	0.739	**0.817**	0.721	0.692	0.745	0.751	0.744	0.742	<u>0.806</u>
CutDiff	0.743	0.711	0.739	0.730	0.744	0.747	0.741	<u>0.777</u>	**0.811**
CutPaste	0.788	0.841	0.694	0.756	0.818	<u>0.862</u>	0.830	0.850	**0.884**
f_{aug}	Avg	Rand	Base	MMD	STD	MC	SEL	HITS	**DSV**
CutOut	7.33	6.10	6.62	6.93	6.29	6.50	7.10	<u>5.43</u>	**3.79**
CutAvg	7.00	<u>5.02</u>	7.64	8.36	5.52	5.48	5.98	5.60	**4.19**
CutDiff	6.43	7.24	6.45	7.38	6.00	<u>5.64</u>	6.24	6.21	**3.60**
CutPaste	7.67	6.29	8.67	7.21	5.60	**4.33**	5.17	4.64	<u>4.57</u>

where $\text{std}(\mathcal{A}) = \sqrt{|\mathcal{A}|^{-1} \sum_{a \in \mathcal{A}} (a - \text{mean}(\mathcal{A}))}$ is the standard variation of a set, and μ_{trn} is the mean vector of $\mathcal{Z}_{\text{trn}}$. One notable difference from Eq. (4) is that only the mean μ_{trn} is used in the numerator, instead of whole $\mathcal{Z}_{\text{trn}}$, based on the observation that $\mathcal{Z}_{\text{trn}}$ is usually densely clustered as a result of training.

Intuitively, $\mathcal{L}_{\text{sep}}$ measures how much $\mathcal{Z}_{\text{test}}$ is scattered along the direction of $\mathbf{z}_{\text{aug}} - \mu_{\text{trn}}$. The amount of scatteredness is directly related to the value of h_s, since we assume by convention that $\mathcal{Z}^{(n)}_{\text{test}}$ is close to $\mathcal{Z}_{\text{trn}}$. In Lemma 2, we show that $\mathcal{L}_{\text{sep}}$ is a linear function of h_s if some constraints are met, and its optimum is 1/2 in the ideal case, which corresponds to $h_s = 1$, if $\bar{\sigma}_{\text{test}} \ll \|\mathbf{z}_{\text{aug}} - \mathbf{z}_{\text{trn}}\|$.

Lemma 2. *We assume that* $\mathcal{Z}_{\text{trn}} = \{\mathbf{z}_{\text{trn}}\}$, $\mathcal{Z}_{\text{aug}} = \{\mathbf{z}_{\text{aug}}\}$, *and* $\mathbf{z}^{(n)}_{\text{test}} = \mathbf{z}_{\text{trn}}$ *for all* $\mathbf{z}^{(n)}_{\text{test}} \in \mathcal{Z}^{(n)}_{\text{test}}$. *Let* $\gamma = |\mathcal{Z}^{(a)}_{\text{test}}|/|\mathcal{Z}_{\text{test}}|$, *and* $\bar{\sigma}_{\text{test}}$ *be the standard deviation of the projected norms* $\mathcal{Z}^{(p)}_{\text{test}} = \{\text{proj}(\mathbf{z}_{\text{trn}}, \mathbf{z}_{\text{aug}}, \mathbf{z}) \mid \mathbf{z} \in \mathcal{Z}^{(a)}_{\text{test}}\}$. *Then, the separability surrogate loss* $\mathcal{L}_{\text{sep}}$ *is rewritten as a function of* h_s *as follows:*

$$\mathcal{L}_{\text{sep}}(\mathcal{Z}_{\text{trn}}, \mathcal{Z}_{\text{aug}}, \mathcal{Z}_{\text{test}}) = \sqrt{\gamma(1-\gamma)} h_s + \frac{\sqrt{\gamma}\bar{\sigma}_{\text{test}}}{\|\mathbf{z}_{\text{aug}} - \mathbf{z}_{\text{trn}}\|}.$$

Proof. The proof is in Appendix A.2. □

4 Experiments

We answer the following questions through experiments on real datasets:

Q1. **Performance.** Are the models selected by DSV better than those selected by baseline measures for unsupervised model selection? Is the improvement statistically significant across different tasks and datasets?

Q2. **Ablation study.** Are the two main components of DSV for the discordance and separability, respectively, meaningful to performance? How do they complement each other across different augmentation functions and tasks?

Q3. **Case studies.** How does DSV work on individual cases with respect to the distribution of embedding vectors or anomaly scores?

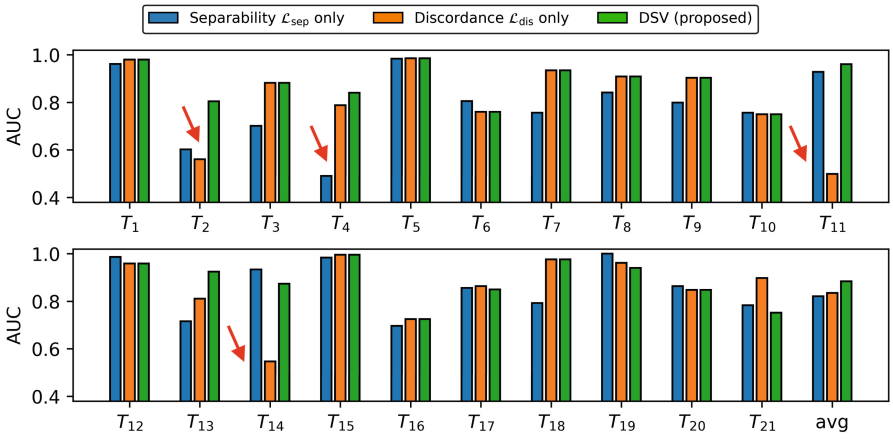
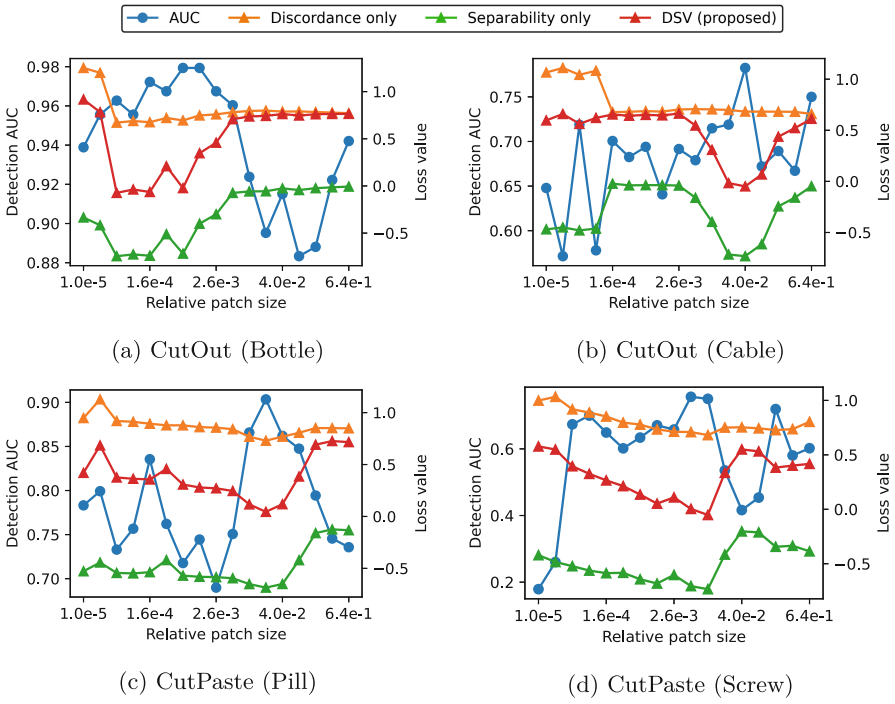


Fig. 3. Ablation study to compare $\mathcal{L}_{\text{dis}}$, $\mathcal{L}_{\text{sep}}$, and $\mathcal{L}_{\text{DSV}}$ on 21 different tasks and on average when f_{aug} = CutPaste. DSV shows a dramatic improvement in a few cases, such as tasks T_2 (both fail), T_4 ($\mathcal{L}_{\text{sep}}$ fails), T_{11} and T_{14} ($\mathcal{L}_{\text{dis}}$ fails).

4.1 Experimental Settings

Datasets. We include two datasets for anomaly detection in natural images: MVTec AD [3] and MPDD [10], which contain 21 different tasks in total. MVTec AD mimics real-world industrial inspection scenarios and contains 15 different tasks: five unique textures and ten unique objects from different domains. MPDD focuses on defect detection during painted metal parts fabrication and contains 6 different object types with a non-homogeneous background. The evaluation is done by AUC (the area under the ROC curve) scores on test data.

Detector Models. We use a classifier-based anomaly detector model used in a previous work [13], which first learns data embeddings and then computes anomaly scores on the space. The model structure is based on ResNet18 [9]. All model hyperparameters are set to the default setting, except for the number of training updates, which we changed for MPDD since the model converged much faster due to the smaller data size; we set the number of updates to 10,000 in MVTec AD, while to 1,000 in MPDD.



(a) CutOut (Bottle)

(b) CutOut (Cable)

(c) CutPaste (Pill)

(d) CutPaste (Screw)

Fig. 4. The AUC and loss values $\mathcal{L}_{\text{dis}}$, $\mathcal{L}_{\text{sep}}$, and $\mathcal{L}_{\text{DSV}}$ with CutOut or CutPaste as f_{aug}. We preprocessed $\mathcal{L}_{\text{sep}}$ so that it can be directly added to $\mathcal{L}_{\text{dis}}$ for creating $\mathcal{L}_{\text{DSV}}$. We have two main observations from the figures. First, $\mathcal{L}_{\text{DSV}}$ is negatively correlated with the actual AUC. Second, $\mathcal{L}_{\text{sep}}$ and $\mathcal{L}_{\text{dis}}$ work in a complementary way, which is shown especially well on (a) and (b).

Augmentation Functions. We use four different augmentation functions in experiments: CutOut [5], CutAvg, CutDiff, and CutPaste [13]. CutOut replaces a random patch from an image with black pixels. CutAvg is similar to CutOut, but it replaces a patch with the average color of the patch, instead of the black. CutDiff is a smooth version of CutOut, and it makes a smooth boundary when selecting a patch. The resulting image has the black at the center of the original position of the patch, and it becomes brighter as it goes close to the boundary. CutPaste copies a patch and pastes it into a random location of the image.

We use the patch size as the target augmentation hyperparameter to search for all these functions, since it directly controls the amount of modification by f_{aug}. We consider 17 settings in the range from 10^{-5} to 0.64 in the log scale. For example, 0.1 represents we select a patch whose size is 10% of the image.

Baselines. We compare our DSV with eight baseline methods for unsupervised model selection. *Average* is the simplest one, which is to take the average performance of all settings we consider. *Random* means we change the hyperparameter for each inference during training and test. *Base* is to use the distance

$\mathcal{L}_{\text{dis}}(\mathcal{Z}_{\text{trn}}, \mathcal{Z}_{\text{aug}}, \mathcal{Z}_{\text{test}})$ as the simplest approximation of $\mathcal{L}_{\text{ali}}$. *MMD* replaces the distance function in Base with the maximum mean discrepancy [21]. *STD* measures standard deviation of the all-pair distances between $\mathcal{Z}_{\text{trn}}$ and $\mathcal{Z}_{\text{test}}$.

MC, SEL, and HITS were proposed in a previous work [15] for unsupervised outlier model selection (see §2.3). They are top-performing baselines based on internal performance measures. MC [14,15] combines different models based on outlier score similarities, assuming that good models have similar outputs as the optimal model, and thus are close to each other. HITS uses the HITS algorithm originally designed for web graphs [11] to compute the importance of each model. SELECT (SEL in short) originates from model ensembles [18,28], and calculates the similarity between the output of each model and the "pseudo ground truth" which is initialized to the average of all candidate models.

4.2 Detection Performance (Q1)

Table 1 shows the average AUC and rank of various methods on 21 different tasks. Due to the lack of space, we include the full results on individual tasks in the supplementary material. DSV shows the best performance on 6 out of the 8 cases, and the second-best on the remaining two cases. MC and HITS perform well compared to the other baselines, but their performances are not consistent across different augmentation functions and tasks.

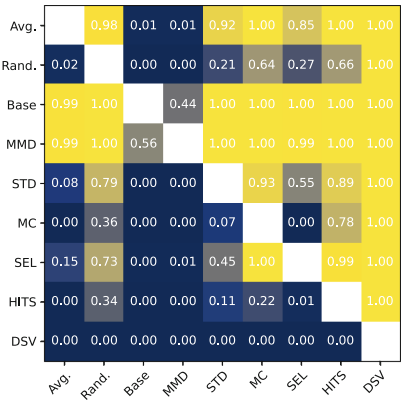
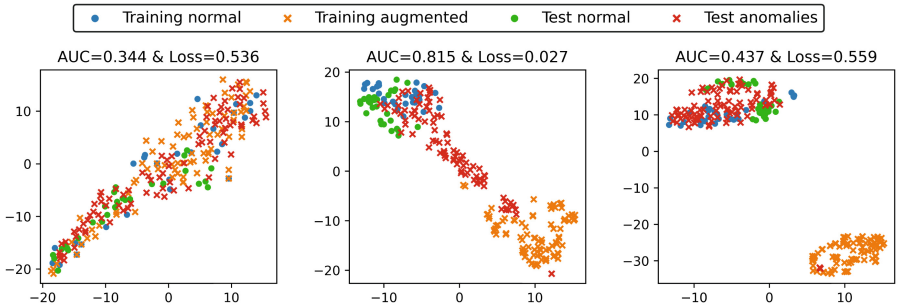
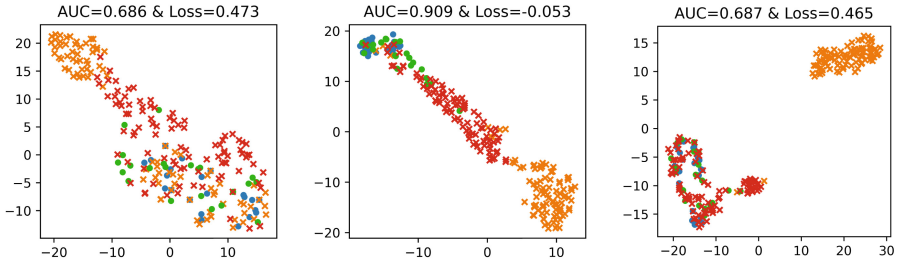


Fig. 5. Wilcoxon signed rank test for all pairs of approaches. DSV is superior to all other approaches with p-values smaller than 0.001.

In Fig. 5, we perform the Wilcoxon signed rank test [8] to check if the differences between models are statistically significant. Each number in the (i, j)-th cell represents the p-value comparing models i and j, and it represents model i is significantly better than model j if the p-value is smaller than 0.05. DSV is significantly better than all of the other approaches in the figure, demonstrating its superiority in unsupervised model selection.

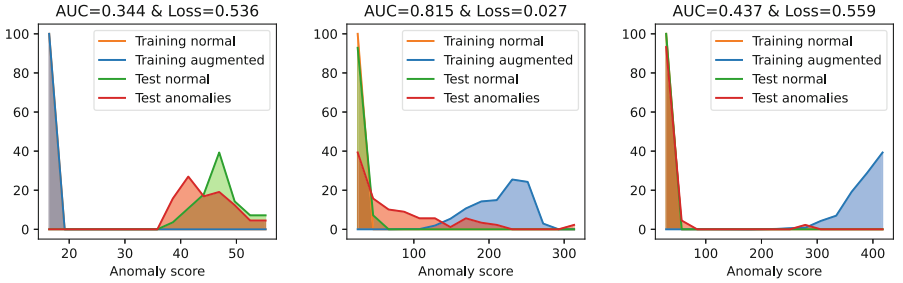


(a) CutOut (0.00004) (b) CutOut (0.00128) (c) CutOut (0.04)



(d) CutPaste (0.00016) (e) CutPaste (0.02) (f) CutPaste (0.32)

Fig. 6. t-SNE visualizations of embeddings in (top) f_{aug} = CutOut and (bottom) f_{aug} = CutPaste, where values in parentheses represent different HPs. $\mathcal{L}_{\text{DSV}}$ is the smallest in (b) and (e), where the anomalies are in between $\mathcal{Z}_{\text{trn}}$ and $\mathcal{Z}_{\text{aug}}$. Detection fails in (a), (c) & (d), (f), showing larger $\mathcal{L}_{\text{DSV}}$ than in (b) & (e), resp.



(a) CutOut (0.00004) (b) CutOut (0.00128) (c) CutOut (0.04)

Fig. 7. Anomaly scores for the three different HPs of f_{aug} = CutOut in Fig. 6. The distributions of embeddings are clearly observed also in the scores: (a) No separation in test data, (b) reasonable separation with as high AUC as 0.815, and (c) drastic separation between augmented points and all other sets.

4.3 Ablation Studies (Q2)

We perform an ablation study in Fig. 3, comparing $\mathcal{L}_{\mathrm{DSV}}$ with its two surrogate losses $\mathcal{L}_{\mathrm{dis}}$ and $\mathcal{L}_{\mathrm{sep}}$ on $f_{\mathrm{aug}} = \mathrm{CutPaste}$. The difference between the three models is more significant in individual cases, rather than on average, as denoted by the red arrows in the figure. This is because each of $\mathcal{L}_{\mathrm{dis}}$ and $\mathcal{L}_{\mathrm{sep}}$ is incomplete by its design. For example, h_d surpasses h_s on average, but it shows some dramatic failure cases as in T_{11} and T_{14}. Our proposed $\mathcal{L}_{\mathrm{DSV}}$ avoids such failures, achieving the best performance by effectively combining the two terms.

The complementary roles of the two losses is also shown in Fig. 4, where we draw actual AUC and three different losses together for various combinations of f_{aug} and tasks. Overall, the value of $\mathcal{L}_{\mathrm{DSV}}$ is negatively correlated with the true AUC, which is exactly the purpose of introducing $\mathcal{L}_{\mathrm{DSV}}$ for unsupervised model selection. In detail, we observe complementary interactions between $\mathcal{L}_{\mathrm{dis}}$ and $\mathcal{L}_{\mathrm{sep}}$ from the figures; for example, in Fig. 4a, $\mathcal{L}_{\mathrm{sep}}$ makes the overall loss decrease when AUC peaks the top, although $\mathcal{L}_{\mathrm{dis}}$ makes only negligible changes. In Fig. 4b, in contrast, the two losses change drastically in small patch sizes, while their sums remain similar, allowing us to avoid HPs with low AUC.

4.4 Case Studies (Q3)

In Fig. 6, we visualize the embeddings when $f_{\mathrm{aug}} = \mathrm{CutOut}$ (the task is Carpet) and $f_{\mathrm{aug}} = \mathrm{CutPaste}$ (the task is Metal Nut). In Figs. 6b and 6e, which show the smallest $\mathcal{L}_{\mathrm{DSV}}$, test anomalies $\mathcal{Z}_{\mathrm{test}}^{(a)}$ are scattered in between $\mathcal{Z}_{\mathrm{trn}}$ and $\mathcal{Z}_{\mathrm{aug}}$. Although some of $\mathcal{Z}_{\mathrm{test}}^{(a)}$ are mixed with $\mathcal{Z}_{\mathrm{test}}^{(n)}$ in Fig. 6b, the AUC is as high as 0.815. On the other hand, in Figs. 6a and 6c, the AUC is lower than even 0.5, while $\mathcal{L}_{\mathrm{DSV}}$ is large. In Fig. 6a, $\mathcal{Z}_{\mathrm{test}}^{(n)}$ and $\mathcal{Z}_{\mathrm{test}}^{(a)}$ are mixed completely, since the amount of modification through augmentation is too small. In Fig. 6c, $\mathcal{Z}_{\mathrm{aug}}$ are separated from all other sets, due to the drastic augmentation. Figures 6d and 6f show similar patterns, although the AUC is generally higher than in CutOut.

In Fig. 7, we visualize the anomaly scores generated by our detector model, following the same scenarios as in Fig. 6 when $f_{\mathrm{aug}} = \mathrm{CutOut}$. Since the detector model in our experiments computes an anomaly score based on the likelihood of a Gaussian mixture model in the embedding space, the scores are related to the actual distances. The scores represent the difference between different HPs well, leading to the observations consistent with the t-SNE visualization.

5 Conclusion

There has been a recent surge of self-supervised learning methods for anomaly detection (SSAD), but how to systematically choose the augmentation hyperparameters here remains vastly understudied. To address this, we introduce DSV, an unsupervised validation loss for selecting optimal SSAD models with effective augmentation hyperparameters. The main idea is to maximize the alignment between augmentation and unknown anomalies with surrogate losses that estimate the discordance and separability of test data. Our experiments demonstrate that DSV outperforms a broad range of baselines. Future work involves extending it to incorporate other distance measures such as the Chebyshev distance.

Acknowledgments. This work is partially sponsored by PwC Risk and Regulatory Services Innovation Center at Carnegie Mellon University. Any conclusions expressed in this material are those of the author and do not necessarily reflect the views, expressed or implied, of the funding parties.

A Proofs of Lemmas

A.1 Proof of Lemma 1

Proof. Let $\hat{\sigma} = \sigma + \epsilon$. We rewrite the numerator of $\mathcal{L}_{\text{dis}}$ based on the definition of h and Assumption 1.

$$d(\mathcal{Z}_{\text{trn}} \cup \mathcal{Z}_{\text{aug}}, \mathcal{Z}_{\text{test}}) = c_1\hat{\sigma} + c_2((1+h)d(\mathcal{Z}_{\text{trn}}, \mathcal{Z}_{\text{aug}}) - d(\mathcal{Z}_{\text{aug}}, \mathcal{Z}^{(a)}_{\text{test}})) + c_3 d(\mathcal{Z}_{\text{aug}}, \mathcal{Z}^{(n)}_{\text{test}}) + c_4 d(\mathcal{Z}_{\text{aug}}, \mathcal{Z}^{(a)}_{\text{test}})$$

Then, we derive the lower bound as follows:

$$\begin{aligned} & d(\mathcal{Z}_{\text{trn}} \cup \mathcal{Z}_{\text{aug}}, \mathcal{Z}_{\text{test}}) \\ & \quad \geq c_2((1+h)d(\mathcal{Z}_{\text{trn}}, \mathcal{Z}_{\text{aug}}) - d(\mathcal{Z}_{\text{aug}}, \mathcal{Z}^{(a)}_{\text{test}})) \\ & \qquad + c_3 d(\mathcal{Z}_{\text{trn}}, \mathcal{Z}_{\text{aug}}) + c_4 d(\mathcal{Z}_{\text{aug}}, \mathcal{Z}^{(a)}_{\text{test}}) + (c_1 - c_3)\hat{\sigma} \\ & \quad = (c_4 - c_2)d(\mathcal{Z}_{\text{aug}}, \mathcal{Z}^{(a)}_{\text{test}}) + (c_2 + c_2 h + c_3)d(\mathcal{Z}_{\text{trn}}, \mathcal{Z}_{\text{aug}}) + (c_1 - c_3)\hat{\sigma} \end{aligned}$$

Similarly, the upper bound is given as follows:

$$\begin{aligned} & d(\mathcal{Z}_{\text{trn}} \cup \mathcal{Z}_{\text{aug}}, \mathcal{Z}_{\text{test}}) \\ & \quad \leq c_2((1+h)d(\mathcal{Z}_{\text{trn}}, \mathcal{Z}_{\text{aug}}) - d(\mathcal{Z}_{\text{aug}}, \mathcal{Z}^{(a)}_{\text{test}})) \\ & \qquad + c_3 d(\mathcal{Z}_{\text{trn}}, \mathcal{Z}_{\text{aug}}) + c_4 d(\mathcal{Z}_{\text{aug}}, \mathcal{Z}^{(a)}_{\text{test}}) + (c_1 + c_3)\hat{\sigma} \\ & \quad = (c_4 - c_2)d(\mathcal{Z}_{\text{aug}}, \mathcal{Z}^{(a)}_{\text{test}}) + (c_2 + c_2 h + c_3)d(\mathcal{Z}_{\text{trn}}, \mathcal{Z}_{\text{aug}}) + (c_1 + c_3)\hat{\sigma} \end{aligned}$$

If we apply the assumption $|\mathcal{Z}_{\text{trn}}| = |\mathcal{Z}_{\text{aug}}|$, which results in $c_2 = c_4$, the first term from both bounds disappears. We get the inequalities in the lemma by dividing both bounds by $d(\mathcal{Z}_{\text{trn}}, \mathcal{Z}_{\text{aug}})$. □

A.2 Proof of Lemma 2

Proof. Let $\mu_{\text{test}} = \text{mean}(\{\text{proj}(\mathbf{z}_{\text{trn}}, \mathbf{z}_{\text{aug}}, \mathbf{z}) \mid \mathbf{z} \in \mathcal{Z}^{(a)}_{\text{test}}\})$ be the average of projected norms. We first rewrite h_s as follows:

$$\begin{aligned} h_s &= \frac{\sum_{\mathbf{z}_{\text{trn}}, \mathbf{z}_{\text{aug}}, \mathbf{z}^{(a)}_{\text{test}} \in \mathcal{Z}_{\text{trn}}, \mathcal{Z}_{\text{aug}}, \mathcal{Z}^{(a)}_{\text{test}}} \text{proj}(\mathbf{z}_{\text{trn}}, \mathbf{z}_{\text{aug}}, \mathbf{z}^{(a)}_{\text{test}})}{d(\mathcal{Z}_{\text{trn}}, \mathcal{Z}_{\text{aug}})|\mathcal{Z}_{\text{trn}}||\mathcal{Z}_{\text{aug}}||\mathcal{Z}^{(a)}_{\text{test}}|} \\ &= \frac{\sum_{\mathbf{z}^{(a)}_{\text{test}} \in \mathcal{Z}^{(a)}_{\text{test}}} \text{proj}(\mathbf{z}_{\text{trn}}, \mathbf{z}_{\text{aug}}, \mathbf{z}^{(a)}_{\text{test}})}{\|\mathbf{z}_{\text{aug}} - \mathbf{z}_{\text{trn}}\||\mathcal{Z}^{(a)}_{\text{test}}|} \\ &= \frac{|\mathcal{Z}^{(a)}_{\text{test}}|\mu_{\text{test}}}{\|\mathbf{z}_{\text{aug}} - \mathbf{z}_{\text{trn}}\||\mathcal{Z}^{(a)}_{\text{test}}|} = \frac{\mu_{\text{test}}}{\|\mathbf{z}_{\text{aug}} - \mathbf{z}_{\text{trn}}\|} \end{aligned}$$

We rewrite the *squared* numerator of $\mathcal{L}_{\text{sep}}$:

$$\begin{aligned}
&\text{std}^2(\{\text{proj}(\mu_{\text{trn}}, \mathbf{z}_{\text{aug}}, \mathbf{z}_{\text{test}}) \mid \mathbf{z}_{\text{aug}}, \mathbf{z}_{\text{test}} \in \mathcal{Z}_{\text{aug}}, \mathcal{Z}_{\text{test}}\}) \\
&\quad = \text{std}^2(\{\text{proj}(\mu_{\text{trn}}, \mathbf{z}_{\text{aug}}, \mathbf{z}_{\text{test}}) \mid \mathbf{z}_{\text{test}} \in \mathcal{Z}_{\text{test}}\}) \\
&\quad = \frac{1}{|\mathcal{Z}_{\text{test}}|} \sum_{\mathbf{z}_{\text{test}}} (\text{proj}(\mu_{\text{trn}}, \mathbf{z}_{\text{aug}}, \mathbf{z}_{\text{test}}) - \gamma\mu_{\text{test}})^2 \\
&\quad = \frac{1}{|\mathcal{Z}_{\text{test}}|} \left(|\mathcal{Z}_{\text{test}}^{(n)}| \gamma^2 \mu_{\text{test}}^2 + |\mathcal{Z}_{\text{test}}^{(a)}| (\bar{\sigma}_{\text{test}}^2 + (1-\gamma)^2 \mu_{\text{test}}^2) \right) \\
&\quad = (1-\gamma)\gamma^2 \mu_{\text{test}}^2 + \gamma(\bar{\sigma}_{\text{test}}^2 + (1-\gamma)^2 \mu_{\text{test}}^2) \\
&\quad = \gamma(1-\gamma)\mu_{\text{test}}^2 + \gamma\bar{\sigma}_{\text{test}}^2.
\end{aligned}$$

Then, $\mathcal{L}_{\text{sep}}$ is rewritten as follows:

$$\mathcal{L}_{\text{sep}} = \frac{\sqrt{\gamma(1-\gamma)\mu_{\text{test}}^2 + \gamma\bar{\sigma}_{\text{test}}^2}}{d(\mathcal{Z}_{\text{trn}}, \mathcal{Z}_{\text{aug}})} = \sqrt{\gamma(1-\gamma)} h_s + \frac{\sqrt{\gamma}\bar{\sigma}_{\text{test}}}{\|\mathbf{z}_{\text{aug}} - \mathbf{z}_{\text{trn}}\|},$$

which is the equation in the lemma. □

References

1. Baevski, A., Hsu, W.N., Xu, Q., Babu, A., Gu, J., Auli, M.: Data2vec: a general framework for self-supervised learning in speech, vision and language. In: International Conference on Machine Learning, pp. 1298–1312. PMLR (2022)
2. Bergman, L., Hoshen, Y.: Classification-based anomaly detection for general data. In: ICLR (2020)
3. Bergmann, P., Fauser, M., Sattlegger, D., Steger, C.: Mvtec AD - A comprehensive real-world dataset for unsupervised anomaly detection. In: CVPR (2019)
4. Chen, X., Xie, S., He, K.: An empirical study of training self-supervised vision transformers. In: ICCV (2021)
5. Devries, T., Taylor, G.W.: Improved regularization of convolutional neural networks with cutout. CoRR abs/1708.04552 (2017)
6. Elnaggar, A.: Prottrans: toward understanding the language of life through self-supervised learning. IEEE Trans. Pattern Anal. Mach. Intell. **44**(10), 7112–7127 (2021)
7. Golan, I., El-Yaniv, R.: Deep anomaly detection using geometric transformations. In: NeurIPS (2018)
8. Groggel, D.J.: Practical nonparametric statistics. Technometrics **42**(3), 317–318 (2000)
9. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR (2016)
10. Jezek, S., Jonak, M., Burget, R., Dvorak, P., Skotak, M.: Deep learning-based defect detection of metal parts: evaluating current methods in complex conditions. In: ICUMT (2021)
11. Kleinberg, J.M.: Authoritative sources in a hyperlinked environment. J. ACM **46**(5), 604–632 (1999)
12. Kolesnikov, A., Zhai, X., Beyer, L.: Revisiting self-supervised visual representation learning. In: CVPR (2019)

13. Li, C., Sohn, K., Yoon, J., Pfister, T.: Cutpaste: Self-supervised learning for anomaly detection and localization. In: CVPR (2021)
14. Lin, Z., Thekumparampil, K.K., Fanti, G., Oh, S.: InfoGAN-CR and modelcentrality: Self-supervised model training and selection for disentangling GANs. In: ICML (2020)
15. Ma, M.Q., Zhao, Y., Zhang, X., Akoglu, L.: The need for unsupervised outlier model selection: a review and evaluation of internal evaluation strategies. ACM SIGKDD Explor. Newslett. **25**(1) (2023)
16. MacKay, M., Vicol, P., Lorraine, J., Duvenaud, D., Grosse, R.: Self-tuning networks: Bilevel optimization of hyperparameters using structured best-response functions. arXiv preprint arXiv:1903.03088 (2019)
17. Qiu, C., Pfrommer, T., Kloft, M., Mandt, S., Rudolph, M.: Neural transformation learning for deep anomaly detection beyond images. In: ICML (2021)
18. Rayana, S., Akoglu, L.: Less is more: Building selective anomaly ensembles. ACM Trans. Knowl. Discov. Data **10**(4), 42:1–42:33 (2016)
19. Sehwag, V., Chiang, M., Mittal, P.: SSD: A unified framework for self-supervised outlier detection. In: ICLR (2021)
20. Shenkar, T., Wolf, L.: Anomaly detection for tabular data with internal contrastive learning. In: ICLR (2022)
21. Smola, A.J., Gretton, A., Borgwardt, K.: Maximum mean discrepancy. In: 13th International Conference, ICONIP, pp. 3–6 (2006)
22. Sohn, K., Li, C., Yoon, J., Jin, M., Pfister, T.: Learning and evaluating representations for deep one-class classification. In: ICLR (2021)
23. Wolpert, D.H., Macready, W.G.: No free lunch theorems for optimization. IEEE Trans. Evol. Comput. **1**(1), 67–82 (1997)
24. Ye, Z., Chen, Y., Zheng, H.: Understanding the effect of bias in deep anomaly detection. In: IJCAI (2021)
25. Yoo, J., Zhao, T., Akoglu, L.: Self-supervision is not magic: Understanding data augmentation in image anomaly detection. arXiv (2022)
26. Zhao, Y., Rossi, R., Akoglu, L.: Automatic unsupervised outlier model selection. Adv. Neural. Inf. Process. Syst. **34**, 4489–4502 (2021)
27. Zhao, Y., Zhang, S., Akoglu, L.: Toward unsupervised outlier model selection. In: ICDM, pp. 773–782. IEEE (2022)
28. Zimek, A., Campello, R.J.G.B., Sander, J.: Ensembles for unsupervised outlier detection: challenges and research questions a position paper. SIGKDD Explor. **15**(1), 11–22 (2013)
29. Zoph, B., Cubuk, E.D., Ghiasi, G., Lin, T.Y., Shlens, J., Le, Q.V.: Learning data augmentation strategies for object detection. In: ECCV (2020)