FUN-AD: Fully Unsupervised Learning for Anomaly Detection with Noisy Training Data

Jiin Im[1,*] Yongho Son[2,*] Seok Young Hong[3,†] Je Hyeong Hong[1,2,†]

April 5, 2026

Abstract

While the mainstream research in anomaly detection has mainly followed the one-class classification, practical industrial environments often incur noisy training data due to annotation errors or lack of labels for new or refurbished products. To address these issues, we propose a novel learning-based approach for fully unsupervised anomaly detection with unlabeled and potentially contaminated training data. Our method is motivated by two observations, that i) the pairwise feature distances between the normal samples are on average likely to be smaller than those between the anomaly samples or heterogeneous samples and ii) pairs of features mutually closest to each other are likely to be homogeneous pairs, which hold if the normal data has smaller variance than the anomaly data. Building on the first observation that nearest-neighbor distances can distinguish between confident normal samples and anomalies, we propose a pseudo-labeling strategy using an iteratively reconstructed memory bank (IRMB). The second observation is utilized as a new loss function to promote class-homogeneity between mutually closest pairs thereby reducing the ill-posedness of the task. Experimental results on two public industrial anomaly benchmarks and semantic anomaly examples validate the effectiveness of FUN-AD across different scenarios and anomaly-to-normal ratios. Our code is available at https://github.com/HY-Vision-Lab/FUNAD.

1. Introduction

Anomaly detection refers to the process of detecting events that are rare and interesting, and is an essential application in engineering, science, medicines and finance [6, 23, 40]. In particular, anomaly detection involving visual data has received much attention recently, ranging from defect detection in manufacturing industry [2, 8–11, 20, 22, 25, 28–30, 32, 33, 36, 38, 41, 45–48, 51], lesion detection in medical imaging [37, 39] to violence detection in surveillance [1, 14, 26, 43, 44].

While the task of industrial anomaly detection is relatively well-defined, there are two main issues that raise the difficulty of the problem in practice. First, anomaly samples are rare and consequently difficult to obtain, triggering significant data imbalance between the normal and abnormal classes. Second, anomalies can arise from different causes, leading to a diverse distribution of anomaly samples. Due to these problems, it is a commonly adopted problem setting in industrial anomaly detection [2, 8–10, 20, 22, 28–30, 32, 33, 36, 45, 46, 48] that only the class of normal data is used for training. While these methods are often noted as "unsupervised" approaches, they mostly adopt a form of supervised learning called one-class training since the training data requires correctly labeled normal samples. When training data is contaminated, one-class classification methods which do not separately consider outliers in the training data (e.g. OC-SVM [24] does consider outliers) are vulnerable to contamination and may continue to mistake certain classes of anomalies for normals as they consider anomalies in the training dataset to be normals. Consequently, this problem setting still requires clean data collection, which incurs considerable annotation costs and time. In this study, we explore the possibility of addressing the challenging question "can we train an accurate industrial anomaly detection algorithm without any labeled data?". In real-world scenarios, normal data can easily become outdated due to regular product upgrades or changes in manufacturing processes. Moreover, even

1. Department of Electronic Engineering, Hanyang University, South Korea
2. Department of Artificial Intelligence, Hanyang University, South Korea
3. School of Social Sciences and School of Physical and Mathematical Sciences, Nanyang Technological University, Singapore
*. Equal contribution
†. Corresponding Authors

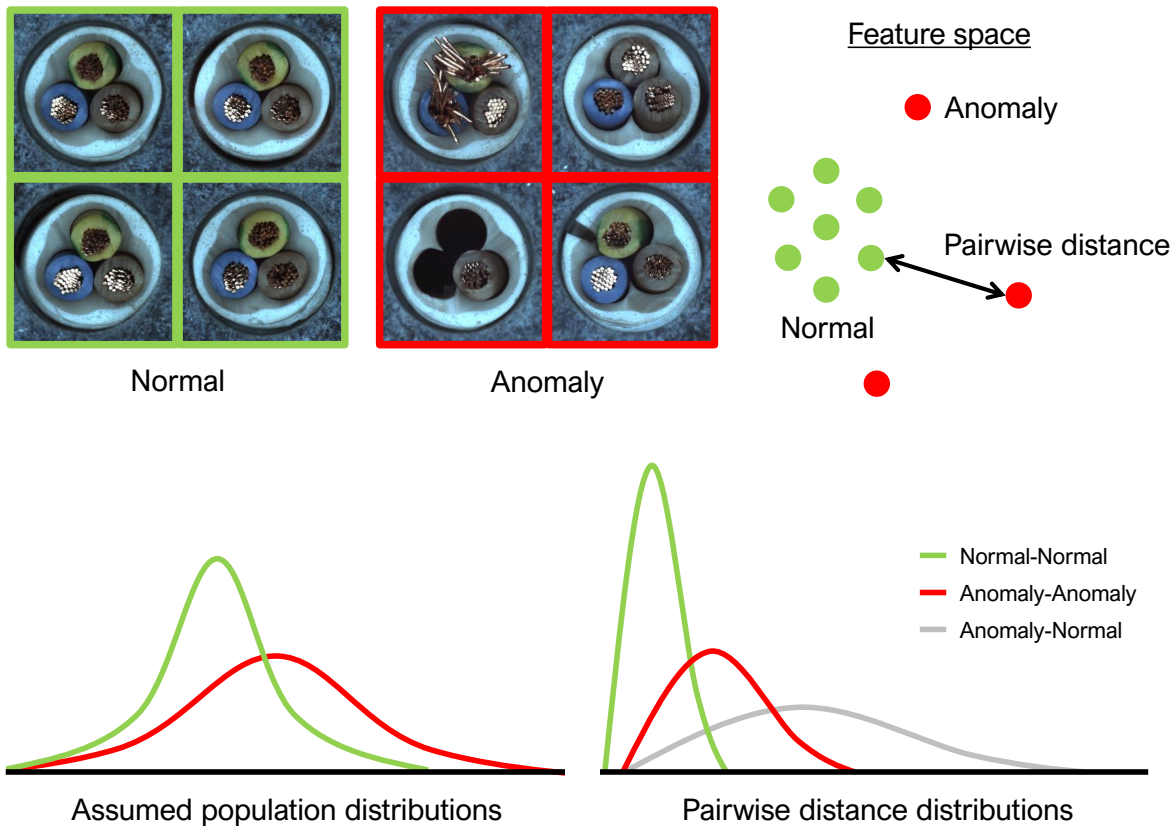


Figure 1. An illustration of our motivation. We assume that the (image-level and patch-level) features from the normal samples (in green) exhibit smaller variance than those from the anomaly samples (in red). In Sec. 3, we analytically show this leads to a homogeneous pair of normal samples being more likely to yield smaller pairwise distance than other types of pairs. In Sec. 3.1, we empirically show that mutually closest pairs are likely to be homogeneous pairs (i.e. mostly normal-normal or anomaly-anomaly).

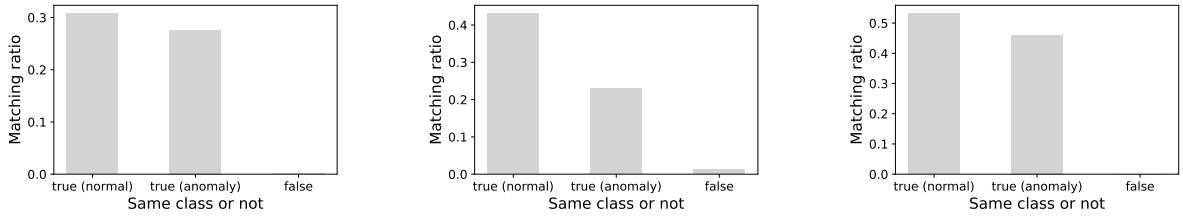
labeled training data can be contaminated with anomalies due to human errors in annotation. All these issues with one-class training would be eradicated in the fully unsupervised setting, providing practical motivation.

While fully unsupervised approaches [7, 21, 25, 38, 42] exist to address the above questions, they mostly focus on eliminating samples pseudo-labeled as anomalies and performing one-class training with the remaining data. However, supervised anomaly detection studies [41, 47] have shown that even limited anomaly information can enhance both anomaly detection and localization performance, while also addressing the issue of small anomalies being overlooked by traditional one-class classification methods.

To this end, we take a different approach towards using anomaly information compared to other fully unsupervised schemes. Our work leverages statistical observations that i) pairs of normal features are likely to be closer together than other types of pairs and ii) mutually closest pairs are likely to be formed by pairs from the same class (homogeneous pairs). We demonstrate that they provide cues to distinguish normal samples from anomalies when the variance of normal data is smaller than that of the anomalies.

We summarize the contributions of our work as follows:

- A previously-untouched statistical analysis of pairwise distance of features which provides a cue to distinguishing confident normal samples from unlabeled data,
- a new pseudo-labeling approach based on iteratively re-constructed memory bank (IRMB) designed to utilize above statistics of pairwise distances,
- a novel *mutual smoothness* loss which reduces the ill-posedness by aligning anomaly scores of mutually closest feature pairs under the validated assumption that they largely belong to the same class, and
- a simple yet effective iterative learning-based framework for fully unsupervised anomaly detection, achieving state-of-the-art (SOTA) performance in anomaly detection and localization across various contaminated settings on public industrial



(a) Synthetic (isotropic Gaussians) (b) Real image features (MVTec) (c) Real patch features (MVTec)

Figure 2. Visualization of the matching ratios for different types of feature pairs. The empirical experimental settings are as in Sec. 1 in [18] for both synthetic and real. For "true (normal)", the matching ratio is the number of samples mutually closest to normal-normal divided by the number of normal samples, and for "true (anomaly)", the matching ratio is the number of samples mutually closest to anomaly-anomaly divided by the total number of anomaly samples. Additionally, "false" represents the number of samples mutually closest to normal-anomaly (or anomaly-normal) divided by the total number of samples (including both normal and anomaly).

datasets (MVTec AD and VisA).

2. Related work

We briefly review studies in anomaly detection that are mostly relevant to our work.

One-class anomaly detection In the one-class classification setting, we assume the training data contains correctly labeled normal samples and no anomalies. While a plethora of different methods exist, these can be largely categorized into i) reconstruction-based methods [32, 45, 46, 48] which learn to reconstruct a normal image from an anomaly sample and detect the anomaly region via difference of images, ii) embedding-based methods [2, 8–10, 22, 28, 29, 33, 36] which measure similarity against normal features extracted from a pretrained network, and iii) self-supervised methods [20, 30, 33] based on generating pseudo-anomalies.

We summarize previous works partially incorporated into our approach, but used differently as detailed in Sec. 4. The first relevant work is PatchCore [28], which utilizes a pretrained feature extractor to obtain normal patch features from training data, subsamples them and stores them in a static memory bank. During inference, query image features are extracted and compared to the memory bank via nearest-neighbor search for anomaly detection.

The second relevant work is SimpleNet [22], which uses self-supervised learning by adding Gaussian noise to normal features to create pseudo anomalies for training alongside normal samples. These pseudo anomalies are generated in feature space and used to train a simple discriminator network to detect anomalies. Nevertheless, these two approaches are designed to work with clean normal samples, which is not robust to noisy training data.

Fully unsupervised anomaly detection More recently, several studies explored fully-unsupervised learning for industrial anomaly detection whereby the training data is unlabeled and may comprise anomalies. Most research [25, 38, 42] attempts to eliminate pseudo-anomalies from the training data and re-deploy one-class anomaly detection [20, 28] on the filtered training set. Xi *et al*. [38] proposed to filter the training data via thresholding based on the value of local outlier factor (LOF) [4] and re-deploy PatchCore [28] on the filtered dataset using reweighted anomaly scores. McIntosh and Albu [25] extracted high-confidence normal patches based on the assumption that normal patch features exhibit high span (large in numbers) and low spread (small diversity), and used them to detect anomaly patches. While both approaches achieve fully unsupervised training, they solely rely on a pretrained feature extractor for constructing a memory bank, so any incorrectly classified anomaly sample can be stuck inside the memory bank and consistently degrade the detection accuracy.

3. Motivations

We illustrate two observations motivating our strategy proposed in Sec. 4. In Sec. 3, we show analytically that the pair of features that are relatively close is most likely to arise from the pair of normal samples. In Sec. 3.1, we empirically demonstrate that the mutually closest feature pairs are highly likely to be derived from the same class. Interestingly, these results only rely on the assumption of smaller variance for the normal data compared to the anomalies, and they do not require the means of two distributions to be different or need anomalies to be scarce.

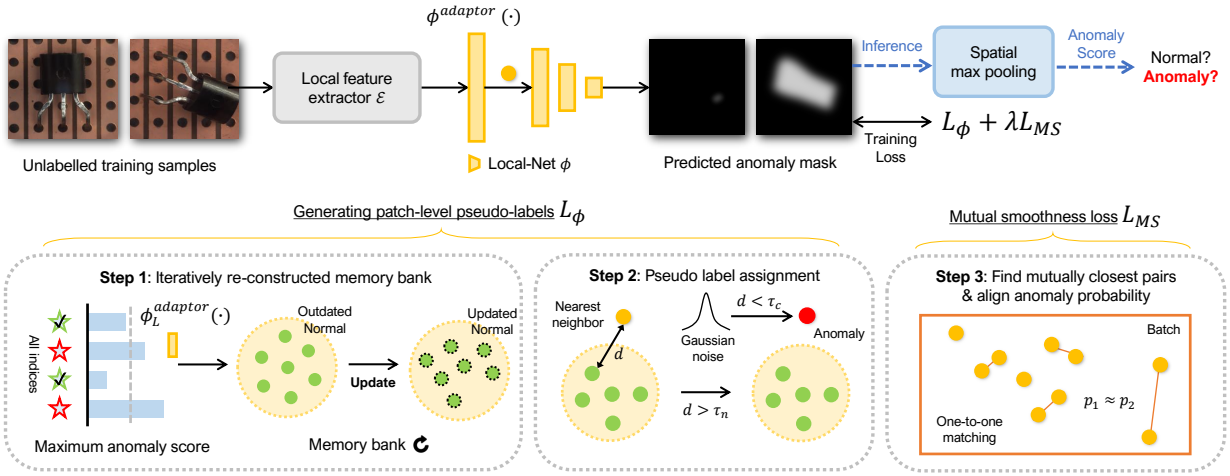


Figure 3. Our framework overview for fully unsupervised anomaly detection. While the framework itself is simple, its constituent components such as patch-level pseudo-label generator and mutual smoothness loss are designed to effectively utilize the observations in Sec. 3.

3.1. Statistical analysis of pairwise distances between features

(1) The Isotropic Gaussian case

For the purpose of intuitive illustration, we present our analysis to an ideal case whereby the normal features and anomaly features follow distinct isotropic Gaussian distributions. Let $\mathbf{x}_N \sim \mathcal{N}(\boldsymbol{\mu}_N, \Sigma_N)$ be the distribution of the normal samples with $\Sigma_N = \sigma_N^2 \mathtt{I}$ and $\mathbf{x}_A \sim \mathcal{N}(\boldsymbol{\mu}_A, \Sigma_A)$ the distribution of the anomaly samples with $\Sigma_A = \sigma_A^2 \mathtt{I}$. Provided that the anomaly samples are more spread out than the normal samples, we acknowledge $\sigma_N < \sigma_A$.

If $\mathbf{x}_1$ and $\mathbf{x}_2$ are samples each drawn from one of the two distributions, then it naturally follows that $\mathbf{u} := \mathbf{x}_1 - \mathbf{x}_2 \sim \mathcal{N}(\boldsymbol{\mu}_{c_1} - \boldsymbol{\mu}_{c_2}, (\sigma_{c_1}^2 + \sigma_{c_2}^2)\mathtt{I})$, where c_1 is the class (N or A) of sample 1 and c_2 is the class of sample 2 respectively. Then, the probability of the distance between $\mathbf{x}_1 \in \mathbb{R}^D$ and $\mathbf{x}_2 \in \mathbb{R}^D$ being less than the threshold $\tau \in \mathbb{R}^+$ can be represented as:

$$P(\|\mathbf{x}_1 - \mathbf{x}_2\|_2 < \tau \mid c_1, c_2) = P(\|\mathbf{u}\|_2^2 < \tau^2 \mid c_1, c_2). \tag{1}$$

If $\mathbf{x}_1$ and $\mathbf{x}_2$ are from the same distribution, then $\mathbf{u} \sim \mathcal{N}(\mathbf{0}, 2\sigma_{c_1}^2 \mathtt{I})$, and thus Eq. (1) becomes

$$\begin{aligned} P\left(\sum_{i=1}^{D} u_i^2 < \tau^2 \mid c_1 = c_2\right) &= P\left(\sum_{i=1}^{D} z_i^2 < \frac{\tau^2}{2\sigma_{c_1}^2}\right) \\ &= F_\chi\left(\frac{\tau^2}{2\sigma_{c_1}^2}\right), \end{aligned} \tag{2}$$

where $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathtt{I})$ and $F_\chi(\cdot)$ is the cumulative distribution function of the chi-squared distribution χ_D^2 with D degrees of freedom. If $\mathbf{x}_1$ and $\mathbf{x}_2$ are drawn from different distributions,

$$\begin{aligned} P\left(\sum_{i=1}^{D} u_i^2 < \tau^2 \mid c_1 \neq c_2\right) &= P\left(\sum_{i=1}^{D} v_i^2 < \frac{\tau^2}{\sigma_N^2 + \sigma_A^2}\right) \\ &= F_{\tilde{\chi}}\left(\frac{\tau^2}{\sigma_N^2 + \sigma_A^2}\right), \end{aligned} \tag{3}$$

where $\mathbf{v} \sim \mathcal{N}((\boldsymbol{\mu}_{c_1} - \boldsymbol{\mu}_{c_2})/\sqrt{\sigma_N^2 + \sigma_A^2}, \mathtt{I})$ and $F_{\tilde{\chi}}(\cdot)$ is the cumulative distribution function of the non-central chi-squared distribution with the non-centrality parameter $\lambda = |\boldsymbol{\mu}_{c_1} - \boldsymbol{\mu}_{c_2}|^2/(\sigma_N^2 + \sigma_A^2)$.

We now analyze the probability in Eq. (1) for different types of features pairs, namely the normal-normal pair, anomaly-anomaly pair and normal-anomaly pair, with $\tau \ll 1$ to simulate close pairs. Since Eq. (1) tends to 0 as $\tau \to 0$ for any type of feature pairs, we instead resort to evaluating the ratio of probabilities between different types of pairs to approximate the

comparative sizes. Comparing the probabilities between the normal-normal pair and the anomaly-anomaly pair using Eq. (2) yields

$$\frac{P(\|\mathbf{x}_1-\mathbf{x}_2\|_2<\tau \mid c_1=c_2=N)}{P(\|\mathbf{x}_1-\mathbf{x}_2\|_2<\tau \mid c_1=c_2=A)}=\frac{F_\chi\left(\tau^2/2\sigma_N^2\right)}{F_\chi\left(\tau^2/2\sigma_A^2\right)}>1 \tag{4}$$

for $\tau \ll 1$ since $\sigma_N^2 < \sigma_A^2$. This means that a pair of normals is *more likely* to be within the distance of τ than a pair of anomalies. We compare the probabilities between the normal-normal pair and the normal-anomaly pair for $\tau \ll 1$, thus

$$\frac{P(\|\mathbf{x}_1-\mathbf{x}_2\|_2<\tau \mid c_1=c_2=N)}{P(\|\mathbf{x}_1-\mathbf{x}_2\|_2<\tau \mid c_1\neq c_2)}=\frac{F_\chi\left(\tau^2/2\sigma_N^2\right)}{F_{\tilde{\chi}}\left(\tau^2/(\sigma_N^2+\sigma_A^2)\right)}. \tag{5}$$

We consider two lower-bound cases to show that Eq. (5) is always greater than 1. First, when the two distributions have the same mean, i.e. $\boldsymbol{\mu}_N = \boldsymbol{\mu}_A$, then $F_{\tilde{\chi}}(\tau^2/(\sigma_N^2+\sigma_A^2))$ approximates to $F_\chi(\tau^2/(\sigma_N^2+\sigma_A^2))$ which is less than $F_\chi(\tau^2/2\sigma_N^2)$ so long as $\sigma_N < \sigma_A$, yielding Eq. (5) to be greater than 1. Second, when the normal and anomaly distributions have substantially different means but similar variances, then $F_{\tilde{\chi}}(\tau^2/(\sigma_N^2+\sigma_A^2)) \approx F_{\tilde{\chi}}(\tau^2/2\sigma_N^2)$ which cannot be larger than $F_\chi(\tau^2/2\sigma_N^2)$ for $\lambda > 0$. In practice, the normal and anomaly data usually have different means and variances to safely go over 1. This implies a pair of normal features is *more likely* to be within the distance of τ than a heterogeneous pair of anomaly and normal features.

(2) The Non-isotropic Gaussian case

While the isotropic assumption yields a chi-squared χ^2 distribution, the same intuition extends to the more general (and practically more relevant) setting where the feature distributions are non-isotropic Gaussian. Specifically, let $\mathbf{x}_N \sim \mathcal{N}(\boldsymbol{\mu}_N, \Sigma_N)$ be the distribution of normal features and $\mathbf{x}_A \sim \mathcal{N}(\boldsymbol{\mu}_A, \Sigma_A)$ the distribution of anomaly features, where Σ_N and Σ_A are symmetric positive definite but not necessarily diagonal nor proportional to $\mathtt{I}$. Given two independent samples $\mathbf{x}_1$ and $\mathbf{x}_2$ drawn from $c_1, c_2 \in \{N, A\}$ respectively, let $\mathbf{u} := \mathbf{x}_1 - \mathbf{x}_2$. Then,

$$\mathbf{u} \sim \mathcal{N}\big(\boldsymbol{\mu}_{c_1}-\boldsymbol{\mu}_{c_2},\ \Sigma_{c_1}+\Sigma_{c_2}\big)\,, \qquad P(\|\mathbf{x}_1-\mathbf{x}_2\|_2<\tau \mid c_1,c_2) \;=\; P(\|\mathbf{u}\|_2^2<\tau^2 \mid c_1,c_2). \tag{6}$$

If $\mathbf{x}_1$ and $\mathbf{x}_2$ are from the same distribution so that $c_1 = c_2 = c$, then $\mathbf{u} \sim \mathcal{N}(\mathbf{0}, 2\Sigma_c)$. Let $\Sigma_c = \mathtt{R}_c\Lambda_c\mathtt{R}_c^\top$ be the eigendecomposition with $\Lambda_c = \mathrm{diag}(\lambda_{c1}, \ldots, \lambda_{cD})$ and $\lambda_{ci} > 0$. Then it follows that $\mathbf{u} = \sqrt{2}\,\mathtt{R}_c\Lambda_c^{1/2}\mathbf{z}$, and

$$\|\mathbf{u}\|_2^2 \;=\; \sum_{i=1}^{D} 2\lambda_{ci} z_i^2. \tag{7}$$

where $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathtt{I})$, as before. Therefore, (6) becomes

$$P(\|\mathbf{x}_1-\mathbf{x}_2\|_2<\tau \mid c_1=c_2=c) \;=\; P\left(\sum_{i=1}^{D} 2\lambda_{ci} z_i^2<\tau^2\right) \;=\; F_{G\chi}\left(\tau^2;\,\mathbf{w}_c\right), \tag{8}$$

where $\mathbf{w}_c := (w_{c1}, w_{c2}, \ldots, w_{cD}) = (2\lambda_{c1}, \ldots, 2\lambda_{cD})$ and $F_{G\chi}(\cdot; \mathbf{w}_c)$ denotes the distribution function of a generalized chi-squared random variable of the form $\sum_i w_{ci}\chi_1^2$ (recovering (2) when $\Sigma_c = \sigma_c^2\mathtt{I}$). If $\mathbf{x}_1$ and $\mathbf{x}_2$ are drawn from different distributions ($c_1 \neq c_2$), define $\delta := \boldsymbol{\mu}_{c_1} - \boldsymbol{\mu}_{c_2}$ and $\Sigma_{NA} := \Sigma_N + \Sigma_A$. Let $\Sigma_{NA} = \mathtt{R}\Lambda\mathtt{R}^\top$ with $\Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_D)$, and set $\mathbf{d} := \mathtt{R}^\top\delta$ and $\kappa_i := d_i/\sqrt{\lambda_i}$. Then for $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathtt{I})$, one can verify

$$\|\mathbf{u}\|_2^2 \;=\; \sum_{i=1}^{D} \lambda_i (z_i+\kappa_i)^2, \tag{9}$$

so equation (6) becomes

$$P\big(\|\mathbf{x}_1-\mathbf{x}_2\|_2<\tau \,|\, c_1\neq c_2\big) \;=\; P\left(\sum_{i=1}^{D} \lambda_i (z_i+\kappa_i)^2<\tau^2\right) \;=\; F_{G\tilde{\chi}}\left(\tau^2;\,\boldsymbol{\lambda},\boldsymbol{\kappa}\right), \tag{10}$$

where $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_D)$, $\boldsymbol{\kappa} = (\kappa_1, \ldots, \kappa_D)$, and $F_{G\tilde{\chi}}(\cdot; \boldsymbol{\lambda}, \boldsymbol{\kappa})$ denotes the distribution function of a generalized non-central chi-squared random variable of the form $\sum_i \lambda_i \tilde{\chi}_1^2(\kappa_i^2)$. We now examine close pairs with $\tau \ll 1$. Since $P(\|\mathbf{x}_1 - \mathbf{x}_2\|_2 < \tau)$

vanishes as $\tau \to 0$ for all pair types, we compare the ratios as before. Using (8), the normal-normal versus anomaly-anomaly ratio is

$$\frac{P(\|\mathbf{x}_1 - \mathbf{x}_2\|_2 < \tau \,|\, c_1 = c_2 = N)}{P(\|\mathbf{x}_1 - \mathbf{x}_2\|_2 < \tau \,|\, c_1 = c_2 = A)} = \frac{F_{G\chi}\big(\tau^2; \mathbf{w}_N\big)}{F_{G\chi}(\tau^2; \mathbf{w}_A)} \;\approx\; \sqrt{\frac{|\Sigma_A|}{|\Sigma_N|}} \;>\; 1, \tag{11}$$

where the approximation follows from the small-ball expansion $P(\|\mathbf{u}\|_2 < \tau) \approx \mathrm{Vol}(B_D(\tau))(2\pi)^{-D/2}|\Sigma|^{-1/2}$ as $\tau \to 0$ for $\mathbf{u} \sim \mathcal{N}(\mathbf{0}, \Sigma)$, and the inequality holds whenever anomalies are more dispersed, e.g. $|\Sigma_N| < |\Sigma_A|$.

Similarly, comparing the normal-normal and normal-anomaly cases gives

$$\frac{P(\|\mathbf{x}_1 - \mathbf{x}_2\|_2 < \tau \,|\, c_1 = c_2 = N)}{P(\|\mathbf{x}_1 - \mathbf{x}_2\|_2 < \tau \,|\, c_1 \neq c_2)} = \frac{F_{G\chi}\big(\tau^2; \mathbf{w}_N\big)}{F_{G\tilde{\chi}}\,(\tau^2; \boldsymbol{\lambda}, \boldsymbol{\kappa})} \tag{12}$$

$$\approx \left(\frac{|\Sigma_N + \Sigma_A|}{|2\Sigma_N|}\right)^{1/2} \exp\left(\frac{1}{2}(\boldsymbol{\mu}_N - \boldsymbol{\mu}_A)^\top (\Sigma_N + \Sigma_A)^{-1}(\boldsymbol{\mu}_N - \boldsymbol{\mu}_A)\right), \tag{13}$$

which becomes larger than 1 when anomalies are more spread out and/or the normal and anomaly means are sufficiently separated.

(3) The Non-Gaussian general cases

The Gaussian assumption in (1) and (2) is mainly used to obtain closed-form distribution function expressions such as chi-squared distributions. However, the key intuition that *close pairs are more likely to be normal-normal* can be justified under much weaker, non-parametric assumptions by analyzing the *small-ball probability* as $\tau \to 0$. To this end, we let the normal and anomaly features follow some arbitrary distributions that are not necessarily Gaussian. We suppose they admit densities f_N and f_A on $\mathbb{R}^D$, respectively.

As before, $\mathbf{x}_1$ and $\mathbf{x}_2$ are two independent samples drawn from $c_1, c_2 \in \{N, A\}$, respectively. We note that

$$p_{c_1c_2}(\tau) := P(\|\mathbf{x}_1 - \mathbf{x}_2\|_2 < \tau \mid c_1, c_2) \;=\; P(\|\mathbf{u}\|_2 < \tau \mid c_1, c_2). \tag{14}$$

$$= \iint \mathbf{1}\{\|\mathbf{x} - \mathbf{y}\|_2 < \tau\}\, f_{c_1}(\mathbf{x})\, f_{c_2}(\mathbf{y})\, d\mathbf{x}\, d\mathbf{y}. \tag{15}$$

Equivalently, letting $f_{\mathbf{u}|c_1c_2}$ denote the density of $\mathbf{u} = \mathbf{x}_1 - \mathbf{x}_2$, we can write

$$p_{c_1c_2}(\tau) \;=\; \int_{\|\mathbf{u}\|_2 < \tau} f_{\mathbf{u}|c_1c_2}(\mathbf{u})\, d\mathbf{u}, \qquad f_{\mathbf{u}|c_1c_2}(\mathbf{u}) \;=\; (f_{c_1} * \tilde{f}_{c_2})(\mathbf{u}), \tag{16}$$

where $\tilde{f}_{c_2}(\mathbf{u}) := f_{c_2}(-\mathbf{u})$ and $*$ denotes convolution. In particular, evaluating at the origin yields

$$f_{\mathbf{u}|c_1c_2}(\mathbf{0}) \;=\; \int f_{c_1}(\mathbf{x})\, f_{c_2}(\mathbf{x})\, d\mathbf{x} \;=\; \langle f_{c_1}, f_{c_2}\rangle, \tag{17}$$

i.e., the inner product between the two class-conditional densities. Assuming a suitable smoothness condition on $f_{\mathbf{u}|c_1c_2}$ at $\mathbf{0}$, we have

$$p_{c_1c_2}(\tau) \;=\; \mathrm{Vol}(B_D(\tau))\, f_{\mathbf{u}|c_1c_2}(\mathbf{0}) \;+\; o(\tau^D), \qquad \tau \to 0, \tag{18}$$

where $\mathrm{Vol}(B_D(\tau)) = \frac{\pi^{D/2}}{\Gamma(D/2+1)}\tau^D$ is the volume of the D-dimensional Euclidean ball of radius τ. Combining (17)-(18), we obtain for $\tau \ll 1$:

$$p_{NN}(\tau) \;\approx\; \mathrm{Vol}(B_D(\tau)) \int f_N(\mathbf{x})^2\, d\mathbf{x} \;=\; \mathrm{Vol}(B_D(\tau))\|f_N\|_2^2, \tag{19}$$

$$p_{AA}(\tau) \;\approx\; \mathrm{Vol}(B_D(\tau)) \int f_A(\mathbf{x})^2\, d\mathbf{x} \;=\; \mathrm{Vol}(B_D(\tau))\|f_A\|_2^2, \tag{20}$$

$$p_{NA}(\tau) \;\approx\; \mathrm{Vol}(B_D(\tau)) \int f_N(\mathbf{x}) f_A(\mathbf{x})\, d\mathbf{x} \;=\; \mathrm{Vol}(B_D(\tau))\langle f_N, f_A\rangle, \tag{21}$$

where $\|f\|_2^2 := \int f(\mathbf{x})^2 d\mathbf{x}$ and $\langle f, g\rangle := \int f(\mathbf{x})g(\mathbf{x})d\mathbf{x}$. Therefore, analogous to (11)-(13), the ratios of close-pair probabilities admit the non-Gaussian approximations

$$\frac{p_{NN}(\tau)}{p_{AA}(\tau)} \approx \frac{\|f_N\|_2^2}{\|f_A\|_2^2}, \qquad \frac{p_{NN}(\tau)}{p_{NA}(\tau)} \approx \frac{\|f_N\|_2^2}{\langle f_N, f_A\rangle}, \qquad \tau \to 0. \tag{22}$$

These expressions provide a clear interpretation: $\|f\|_2$ quantifies the *concentration* (or "peakedness") of a distribution, while $\langle f_N, f_A\rangle$ captures the *overlap* between normals and anomalies. Hence, if anomalies are more spread out so that $\|f_A\|_2 < \|f_N\|_2$, we expect $p_{NN}(\tau) > p_{AA}(\tau)$ for sufficiently small τ; likewise, if f_N and f_A have limited overlap (e.g., due to mean shift or support separation), then $\langle f_N, f_A\rangle$ becomes small, further increasing $p_{NN}(\tau)/p_{NA}(\tau)$.

Remark. We briefly discuss a sufficient "spread-out anomaly" condition. A particularly general way to formalize "anomalies are more dispersed" is to model anomalies via an additive noise as follows:

$$\mathbf{x}_A = \mathbf{x}_N + \boldsymbol{\varepsilon}, \tag{23}$$

where $\boldsymbol{\varepsilon}$ is a random variable with density g, and is independent of $\mathbf{x}_N$. We emphasize that $\boldsymbol{\varepsilon}$ does *not* need to be Gaussian.

We note that $f_A = f_N * g$, and by Young's convolution inequality, see for example, Bogachev (2007),

$$\|f_A\|_2 = \|f_N * g\|_2 \leq \|f_N\|_2 \|g\|_1 = \|f_N\|_2, \tag{24}$$

since $\|g\|_1 = 1$ as g is a density. Moreover, by Cauchy–Schwarz,

$$\langle f_N, f_A\rangle \leq \|f_N\|_2 \|f_A\|_2 \leq \|f_N\|_2^2. \tag{25}$$

Together, (24)–(25) imply that the ratios in (22) are at least 1 (and is strictly larger than 1 unless $f_A \equiv f_N$). This establishes the usual close-pair preference, but without requiring Gaussianity.

3.1. Empirical analysis on mutually closest features

This section is motivated by the seminal work of Zhou *et al*. [50], which leverages the assumption that nearby points are likely to belong to the same class for semi-supervised learning. Similarly, we turn our attention to analyzing the type of feature pairs formed between mutually closest pairs and aim to check if this label-consistency assumption can be applied to our problem. Since the statistical analysis of mutually exclusive pairs is complex, we directly resort to empirical analysis to classify the type (heterogeneous or homogeneous) of these pairs. As in the empirical validation part of Sec. 1 in [18], we perform this analysis on the synthetic data comprising the same isotropic Gaussian distributions and real data from MVTec AD [3].

In the synthetic experiment, we identified the closest sample for each data point and counted the instances where they formed mutually closest pairs. The matching ratio was then calculated as twice the number of unique mutually closest pairs divided by the total number of participating samples. For example, the matching ratio for normal-normal pairs is determined by doubling the number of mutually closest normal-normal pairs and dividing it by the total number of normal samples. As shown in Fig. 2a, nearly all mutually closest pairs were homogeneous.

In the real-world experiment, the same process for calculating the matching ratio was applied separately using image-level features and patch-level features for each sequence in the MVTec AD dataset [3]. The results for both feature levels were averaged to produce Figs. 2b and 2c, which indicate a low proportion of heterogeneous mutually closest pairs. These findings underscore the importance of enforcing class consistency between closest pairs.

4. Proposed method

We describe a learning framework called *FUN-AD* for fully unsupervised industrial anomaly detection, which consists of an anomaly pseudo-labeling method motivated by Sec. 3 and a loss function inspired by Sec. 3.1.

Preliminaries We define the training set as $\mathcal{X} := \{I_i\}_{i=1}^N$, where $I_i \in \mathbb{R}^{H\times W\times 3}$ is the i-th image, N is the number of training samples, and H and W are the image height and width, respectively. We define the j-th patch of I_i as $X_{ij} \in \mathbb{R}^{K\times K\times 3}$, where K is the patch size. FUN-AD comprises two sub-networks: a feature extractor $\mathcal{E}$ and the Local-Net model ϕ for detecting anomalies. I_i is passed through $\mathcal{E}$ to extract the patch-level features $\{\mathbf{f}_{ij}\}_{j=1}^P$.

Algorithm 1 Training procedure for *FUN-AD*

Inputs: Unlabeled training set $\mathcal{X} := \{I_i\}_{i=1}^N$.
Outputs: Weights of the Local-Net ϕ.

for $i = 1, \ldots, N$ **do**
  Extract local patch features $\{\mathbf{f}_{ij}\}_{j=1}^P$ from image I_i.
end for
for $m = 1, \ldots,$ `num_epochs` **do**
  Local-Net Training.
  Divide $\{\mathbf{f}_{ij}\}_{j=1}^P$ for $i = 1, \cdots, N$ into mini-batches $\{\mathcal{B}_n\}$.
  for $n = 1, \ldots,$ `num_iters` **do**
    Construct $\mathcal{M}$ using ϕ, Eq. (6).
    Compute $s_{ij}\ \forall\ \mathbf{f}_{ij} \in \mathcal{B}_n$ using Eq. (7) and Eq. (8).
    Compute hard labels $y_{ij} \leftarrow H(s_{ij} - \tau_n)\ \forall\ \mathbf{f}_{ij} \in \mathcal{B}_n$.
    Find mutually closest feature pairs $(\mathbf{f}_{ij}, \mathbf{f}_{kl})$ within the mini-batch based on Eq. (10).
    Compute mutual smoothness loss using Eq. (9)
    Augment feature $\mathbf{f}_{ij} \leftarrow \mathbf{f}_{ij} + \epsilon\ \forall\ \mathbf{f}_{ij} \in \mathcal{F}$ (See Sec.4.3)
    Perform 1 iteration of nonlinear optimization to minimize $L_\phi + \lambda L_{MS}$.
  end for
end for

4.1. Generating patch-level pseudo-labels from pairwise-distance statistics

From Sec. 3, we note that feature pairs with smaller pairwise distances are more likely to be homogeneous normal pairs, provided that the anomaly features are more spread out than normal features. This observation motivates us to utilize the statistics for pseudo-labeling.

We update the patch feature vector of images classified as normal by ϕ inside the memory bank at each iteration. This implies that even with a randomly constructed (noisy) memory bank containing as many anomalies as normal samples, analyzing the statistics of pairwise distances will allow us to distinguish some confident normal and anomalous samples from the unlabeled training set, providing sufficient supervision to initiate the learning process.

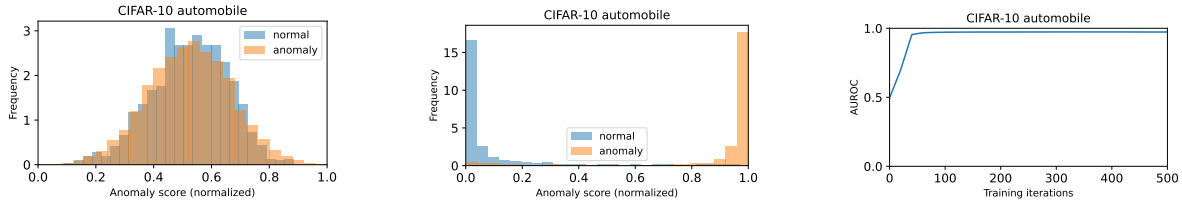
This approach demonstrates that even in the early stages of training, when the memory bank is nearly random, it predominantly consists of normal samples. Since some of the normals in the initial memory bank will pull other normals into memory and push out anomalies, only normals will remain in the memory bank after an iteration. The normal-only memory bank will no longer be noisy and will therefore be better able to distinguish normal from abnormal. Hence, we propose to gradually refine our memory bank features through iteratively re-constructed memory banks and assign pseudo-labels based on pairwise distances.

Iteratively re-constructed memory bank (IRMB) In each iteration, we construct a memory bank comprising features likely derived from normal images. To achieve this, we first estimate the global anomaly score of each image by max-pooling the patch-level anomaly scores of the constituent patches, i.e., $\max_j \phi(\mathbf{f}_{ij})$. Then, we apply min-max normalization to these scores across all training images and use a threshold τ_b to identify a set $\mathcal{P}$ comprising features that are more likely to be normal. Anomaly scores from the local network are normalized, but since they are mostly distributed near 0.5 at the beginning, we perform min-max normalization to distinguish between confident normal and confident abnormal. In terms of equation,

$$\mathcal{P} = \left\{ i \;\middle|\; \frac{\max_j \phi(\mathbf{f}_{ij}) - \min_i \max_j \phi(\mathbf{f}_{ij})}{\max_i \max_j \phi(\mathbf{f}_{ij}) - \min_i \max_j \phi(\mathbf{f}_{ij})} < \tau_b \right\}. \tag{26}$$

Additionally, we sample a random subset $\mathcal{P}' \subset \mathcal{P}$ to reduce computational time. Finally, we construct a memory bank $\mathcal{M} = \{\phi^{adaptor}(\mathbf{f}_{ij}) \mid i \in \mathcal{P}', j = 1, \cdots, P\}$ by storing patch-level features from $\mathcal{P}'$ that have additionally passed through the learnable feature adaptor of the Local-Net ($\phi^{adaptor}$). This allows features from pretrained $\mathcal{E}$ adapt to our anomaly detection task. This feature adaptation along with iteratively reconstructed memory bank allows gradually sharpening of the learning signal.

Pseudo-label assignment In each iteration, we utilize the pairwise distance statistics linked to IRMB for assigning patch-level pseudo labels. We conduct a nearest-neighbor search for each adapted patch feature against the internal features of the



(a) Initial anomaly scores (b) Final anomaly scores (c) AUROC per iteration

Figure 4. Illustration of FUN-AD before and after training on a semantic anomaly detection example. The histograms of anomaly scores from the Local-Net before and after training are shown in (a) and (b) respectively. (c) shows evolution of the image-wise detection accuracy over the iterations.

memory bank $\mathcal{M}$, excluding the feature itself. This exclusion is necessary because the initial Local-Net is random, and not all feature vectors in the memory bank can reliably be considered normal. If a query feature inside the memory bank requires pseudo-labeling, it may continue to be labeled as normal, resulting in persistent incorrect pseudo-labels in the absence of exclusion.

Initially, we define the patch features within the minibatch of the n-th iteration as $\mathcal{B}_n = \{\{\mathbf{f}_{ij}\}_{j=1}^{P}\}_{i=1}^{B}$, where B is the batch size. The nearest-neighbor distance d_{ij} to the features in $\mathcal{M}$ is defined as:

$$d_{ij} = \min\Big(\{\big\|\phi^{adaptor}(\mathbf{f}_{ij}) - \mathbf{m}\big\|_2 : \mathbf{m} \in \mathcal{M} \\ \wedge \big\|\phi^{adaptor}(\mathbf{f}_{ij}) - \mathbf{m}\big\|_2 > 0\}\Big), \tag{27}$$

where $\wedge$ is an *and* operator to remove the case where $\mathbf{m}$ is derived from $\mathbf{f}_{ij}$. This distance is min-max normalized to yield the anomaly score for pseudo labeling (s_{ij}) as:

$$s_{ij} = \frac{d_{ij} - \min_{i,j} d_{ij}}{\max_{i,j} d_{ij} - \min_{i,j} d_{ij}}, \tag{28}$$

which is thresholded to assign the patch-level pseudo label as $y_{ij} = H(s_{ij} - \tau_n)$, where $H(x)$ is a unit step function and τ_n is the threshold below which is classified as normal. Since our goal is to distinguish between outliers and normals, we use a threshold to divide the regions taking advantage of the large spikes in maximum values caused by outliers. This assignment strategy, based on pairwise-distance statistics, is robust to the initially random memory bank and can still provide correct learning signals from the confident normal and anomaly samples. The remaining issues with false positives and false negatives are addressed in Sec. 4.3.

4.2. Mutual-smoothness loss

Following the pseudo-label assignment process, many incorrect pseudo-labels may result from assigning hard labels based on a threshold, potentially leading to inaccurate learning. Therefore, building on the observation from Sec. 3.1 that the mutually closest pairs of features are likely to share the same class, we propose a new *mutual smoothness loss* to align the patch-level anomaly scores of the features forming a mutually-closest pair. For this purpose, we employ the L_1 loss function known for its robustness against noisy labels as outlined in [15], yielding

$$\mathcal{L}_{MS} = \frac{1}{|\Omega|} \sum_{(ij,kl)\in\Omega} |\phi(\mathbf{f}_{ij}) - \phi(\mathbf{f}_{kl})|, \tag{29}$$

where Ω is defined as the unique set of mutually closest pairs such that, for all $(ij, kl) \in \Omega$,

$$\min_{i\neq k, j\neq l} \|\phi^{\text{adaptor}}(\mathbf{f}_{ij}) - \phi^{\text{adaptor}}(\mathbf{f}_{kl})\|_2 \\ = \min_{k\neq i, l\neq j} \|\phi^{\text{adaptor}}(\mathbf{f}_{ij}) - \phi^{\text{adaptor}}(\mathbf{f}_{kl})\|_2. \tag{30}$$

In ambiguous situations where the pseudo-label scores of samples that are mutual nearest-neighbors are close to the threshold, each sample may be assigned a different label. However, since mutually-nearest-neighbor pairs are likely to be in the same class, the anomaly scores between the two samples are made similar to prevent incorrect prediction. The positive effect of this loss is demonstrated in Table 3.

Type		One-class classification					Fully unsupervised		
Dataset	Method	CS-Flow [29]	PaDiM [9]	PatchCore [28]	SimpleNet [22]	RealNet [49]	SoftPatch [38]	InReaCh [25]	FUN-AD (*Ours*)
MVTec AD	*No overlap*	93.8 / -	94.5 / 96.6	98.1 / 94.7	97.7 / 95.3	**99.2** / 96.9	98.3 / 97.3	92.2 / 96.9	**99.2 / 98.6**
	Overlap	65.5 / -	70.1 / 91.4	81.6 / 70.8	59.3 / 52.9	96.8 / 96.0	98.3 / 95.1	92.4 / 97.2	**98.8 / 98.6**
VisA	*No overlap*	82.3 / -	85.8 / 98.3	89.8 / 95.7	90.4 / 96.7	93.5 / 98.7	89.6 / 98.5	83.8 / 97.6	**95.0 / 99.2**
	Overlap	64.3 / -	70.1 / 92.3	82.3 / 81.6	50.4 / 57.9	92.4 / 97.7	89.5 / 96.3	78.5 / 93.6	**95.1 / 99.2**

Table 1. Comparison of quantitative experimental results on the MVTec AD (contaminated) and VisA datasets. The table displays image-wise AUROC (%) / pixel-wise AUROC (%), representing anomaly detection and localization performance, respectively. The best results are in bold and the runner-ups are underlined.

4.3. Training procedure

Subsequently, the Local-Net is trained to minimize the total loss function $\mathcal{L} = L_\phi + \lambda L_{MS}$, which is a weighted sum of the balanced cross-entropy loss L_ϕ and the mutual smoothness loss L_{MS}. L_ϕ is defined as

$$\mathcal{L}_\phi = \frac{1}{|\mathcal{B}_{A,n}|} \sum_{(i,j)\in\mathcal{B}_{A,n}} y_{ij} \ln \phi(\mathbf{f}_{ij}) + \frac{1}{|\mathcal{B}_{N,n}|} \sum_{(i,j)\in\mathcal{B}_{N,n}} (1-y_{ij}) \ln(1-\phi(\mathbf{f}_{ij})) \tag{31}$$

based on the pseudo-labels assigned from Sec. 4.1, and the mutual smoothness loss expressed in Sec. 4.2. $\mathcal{B}_{A,n}$ and $\mathcal{B}_{N,n}$ are the set of samples pseudo-labeled as anomaly and normal respectively in iteration n.

Augmentation of ambiguous features via Gaussian perturbation The hard pseudo-labels from Sec. 4.1 inevitably yield false positives and false negatives. While it is difficult to avoid them completely, we introduce simple feature augmentation to reduce their negative impact. This is achieved by adding Gaussian noise to the set of ambiguous features, which are classified as anomalies, but do not acquire scores above the confident-anomaly threshold of τ_c. This approach mitigates the problem by introducing noise as a perturbation, which prevents the model from incorrectly classifying normal features as anomalies. Our method is partly motivated by [13, 22, 27], which demonstrate that additive Gaussian noise applied to the normal features can generate useful pseudo-anomalies. Above can be expressed as $\mathbf{f}_{ij} \leftarrow \mathbf{f}_{ij} + \epsilon \; \forall \; \mathbf{f}_{ij} \in \mathcal{F}$, where $\mathcal{F} = \{\mathbf{f}_{ij} \mid \tau_n < s_{ij} < \tau_c\}$ is the set of ambiguous anomalies and $\epsilon \sim \mathcal{N}(\mathbf{0}, \Lambda)$ is the Gaussian perturbation with Λ being a diagonal covariance matrix with the elements computed by estimating the element-wise variance of the patch-level features in the mini-batch, i.e. $\mathbf{f}_{ij} \in \mathcal{B}_n$. Finally, the Local-Net is updated by incorporating all of the aforementioned steps. See Algorithm 1 for detailed model training steps.

5. Experimental results and discussions

We compared our method against several baselines using industrial anomaly detection benchmark datasets. We also evaluated the performance when varying the percentage of anomalies in the training dataset, the presence of each module, and the hyperparameters through ablation study.

5.1. Toy example of semantic anomaly detection

We used CIFAR-10 [19] to conduct a toy experiment. The normal class is "automobile", and the outliers consist of the remaining classes in CIFAR-10. Fig. 4a shows the semantic anomaly scores when Local-Net is randomly initialized, and Fig. 4b shows the semantic anomaly scores after Local-Net has been trained by FUN-AD's training process. Fig. 4c illustrates that the AUROC metric shows a clear separation between normal and anomaly, with most of the normal samples remaining in the memory bank as training progresses. For more details, please refer to [18].

5.2. Experiments

Datasets We primarily utilized two widely recognized public benchmarks, MVTec AD [3] and VisA [51]. MVTec AD comprises 15 categories (10 objects, 5 textures), and VisA includes 12 object categories. For MVTec AD, we modified the one-class classification setup by randomly incorporating some of the test set anomalies into the training set at a 1:10 ratio, creating noisy training data contaminated with anomalies. All training samples were stripped of their labels to construct a fully unsupervised setting. In the ***No overlap*** scenario, these relocated anomalies were excluded from evaluation across different anomaly-to-normal ratios. In the ***Overlap*** scenario, the anomalies moved from the test set to the training set were also

Dataset	Method	Anomaly-to-normal ratio					
		0%	1%	3%	5%	10%	20%
MVTec AD [3]	InReach [25]	92.43 / 97.07	92.40 / 97.17	92.40 / 97.11	92.32 / 97.01	92.17 / 96.89	91.28 / 96.94
	SoftPatch [38]	**98.32 / 98.28**	**98.29 / 98.28**	98.38 / 98.27	98.40 / 98.17	98.33 / 97.28	97.73 / 96.87
	FUN-AD	93.33 / 96.19	95.11 / 97.54	95.99 / 97.52	98.45 / 98.43	**99.23 / 98.55**	**98.41** / 98.15
	FUN-AD*	95.63 / 97.51	98.11 / 98.19	**98.51 / 98.37**	**98.85 / 98.49**	98.95 / **98.55**	98.36 / **98.35**
VisA [51]	InReach [25]	83.84 / 97.61	83.96 / 97.65	84.24 / 97.66	83.66 / 97.67	83.72 / 97.58	76.15 / 97.22
	SoftPatch [38]	90.23 / **98.59**	90.08 / 98.56	90.02 / 98.59	89.82 / 98.57	89.69 / 98.58	89.17 / 98.48
	FUN-AD	87.89 / 98.22	89.54 / 98.50	91.63 / 98.70	93.65 / **98.92**	94.57 / **99.13**	94.50 / 99.07
	FUN-AD*	**90.71** / 98.53	**91.85 / 98.61**	**93.35 / 98.90**	**94.25** / 98.88	**94.59 / 99.13**	**94.53 / 99.08**

Table 2. Performance comparison of different fully-unsupervised anomaly detection methods across different anomaly-to-normal ratios on the contaminated MVTec AD and VisA datasets (*no overlap*). * indicates synthetic anomaly data has been utilized for training. The best results are in bold and the runner-ups are underlined.

$\mathcal{L}_{MS}$	ϵ	$\text{AUROC}_{image}(\%)$	$\text{AUROC}_{pixel}(\%)$
		94.95	97.87
✓		96.15	97.87
✓	✓	**98.95**	**98.55**

Table 3. Ablation study of mutual smoothness loss and gaussian noise. ϵ indicates the addition of Gaussian noise to ambiguous samples as described in Sec. 4.3.

used for inference. We also considered ***Overlap*** scenario, as existing fully unsupervised anomaly detection baselines [25, 38] have been evaluated.

Main results In Table 1, *FUN-AD* outperforms previous SOTA methods in both anomaly detection and localization on the contaminated MVTec AD in the ***No overlap*** setting. In the ***Overlap*** setting, our model performs almost as well, unlike the degradation observed in one-class classification models, including fully-unsupervised methods.

Similarly, on VisA, our model exhibits significantly improved results compared to existing models. In the ***Overlap*** setting of VisA, where other one-class classification models show significant performance degradation, our model maintains consistent accuracy. In particular, we show that our method achieves robust performance on the VisA dataset, which contains multiple objects and lacks camera alignment. Fig. 5 shows our method produces sharper boundaries than models that resemble one-class classification.

5.3. Ablation study

In ablation study, we considered the possibility that the training dataset may not be contaminated in real-world scenarios. We emphasize that the synthetic anomalies were generated using a noisy (anomaly-present) dataset, which differs from the approach in [49] that requires clean samples, potentially deteriorating the quality of generated images. We conducted all ablation studies, except those related to semantic anomaly detection, using the training dataset with synthetic anomalies added. Additionally, we used MVTec AD (with 10% contamination) for all ablations except those related to semantic anomaly detection and contamination rate. Additional ablation studies can be found in the supplementary document [18].

Effects of Gaussian noise and mutual smoothness loss Table 3 presents the results of mutual-smoothness loss, and feature augmentation with Gaussian noise. The comparison with and without ϵ in Table 3 demonstrates the effectiveness of the feature augmentation approach introduced in Sec. 4.3. FUN-AD exhibits a significant performance drop without additive Gaussian noise, highlighting the importance of reducing ambiguous anomaly features when providing effective pseudo-anomalies. We demonstrate in Table 3 that mutual smoothness loss performs effectively even in the absence of feature augmentation generated by Gaussian noise, a condition that can lead to many false positives.

Sensitivity to hyperparameters τ_n and τ_c We examined the sensitivity of patch-level pseudo-labeling to hyperparameters τ_n and τ_c. As mentioned in Sec. 4.3, if s_{ij} is larger than τ_c, we consider it as a confident anomaly. On the other hand, if s_{ij} is between τ_n and τ_c, we consider it as an ambiguous situation, where the distinction between normal and anomalous is unclear, and perturb the features by adding Gaussian noise to treat it as an anomaly. The results in Table 4 demonstrate robustness within a range of ± 0.1 from the baseline values of τ_n and τ_c. This suggests that our method is relatively robust to threshold

	τ_n		
τ_c	0.4	0.5	0.6
0.8	98.88 / 98.62	**99.01** / 98.73	98.83 / **98.75**
0.9	98.73 / 98.33	98.95 / 98.55	98.90 / 98.70

Table 4. Ablation study of variations in the anomaly threshold τ_n and confident anomaly threshold τ_c. The table presents the image-wise AUROC (%) / pixel-wise AUROC (%) for assessing anomaly detection and localization performance.

variations, except when τ_c is significantly lower than the default value of 0.9, causing a high rate of false positives, or when τ_n falls substantially below the default value of 0.5, resulting in very few normal samples being labeled as normal.

Effect of different contamination rates Table 2 demonstrates the performance of FUN-AD according to the contamination ratio in the training dataset. Here, "FUN-AD" refers to the results obtained from training with the dataset without synthetic anomalies, while "FUN-AD*" refers to the results from training the FUN-AD framework with synthetic anomalies added at a rate of 5% of the training dataset size. Synthetic anomalies were created from a noisy (anomaly-present) dataset considering a fully unsupervised setting. Given that they contain noisy samples, which could potentially degrade the quality of generated data, their utility in the training process may not always be advantageous.

Since FUN-AD relies on pseudo-labeled anomaly samples for detection, it does not perform as well as other baselines when the training dataset contains very few anomalies. However, when synthetic anomalies are added, the model effectively learns to distinguish between anomalies and normal samples by pseudo-labeling synthetic anomalies in situations where real anomalies are scarce.

6. Conclusion

We have addressed the challenging problem of identifying industrial anomalies without any labeled normal or anomaly data in the fully unsupervised setting whereby the training dataset contains anomalies but the labels are unavailable. Based on the assumption of wider spread for anomalies, we illustrated analytic and empirical motivations for our methodology, namely that normal-normal feature pairs are more likely to form closer feature pairs, and mutually closest pairs are likely to share the same class labels. To incorporate these observations, we presented a novel unsupervised anomaly detection framework, which assigns pseudo-labels based on iteratively re-constructed memory bank and pairwise-distance statistics to achieve robustness to initial noisy labels and allow gradual refinement of the learning signals. We also leveraged the class-consistency of mutually closest features by proposing a new MAE-based mutual smoothness loss for training. Through extensive experimental evaluations, we demonstrated the competitiveness of our approach across different industrial anomaly benchmarks in presence of contaminated training data.

Acknowledgement This work was in part supported by the Technology Innovation Program (1415178807, Development of Industrial Intelligent Technology for Manufacturing, Process, and Logistics) funded by the Ministry of Trade, Industry and Energy (Korea), in part by the National Research Foundation of Korea (NRF) grant funded by the Korean government (No. RS-2023-00302424), in part by Nanyang Technological University (the CoHASS Research Support Grant; No. 025637-00001), and in part by the Institute of Information and communications Technology Planning and Evaluation (IITP) under the artificial intelligence semiconductor support program to nurture the best talents (IITP-2024-RS-2023-00253914) grant funded by the Korean government (MSIT).

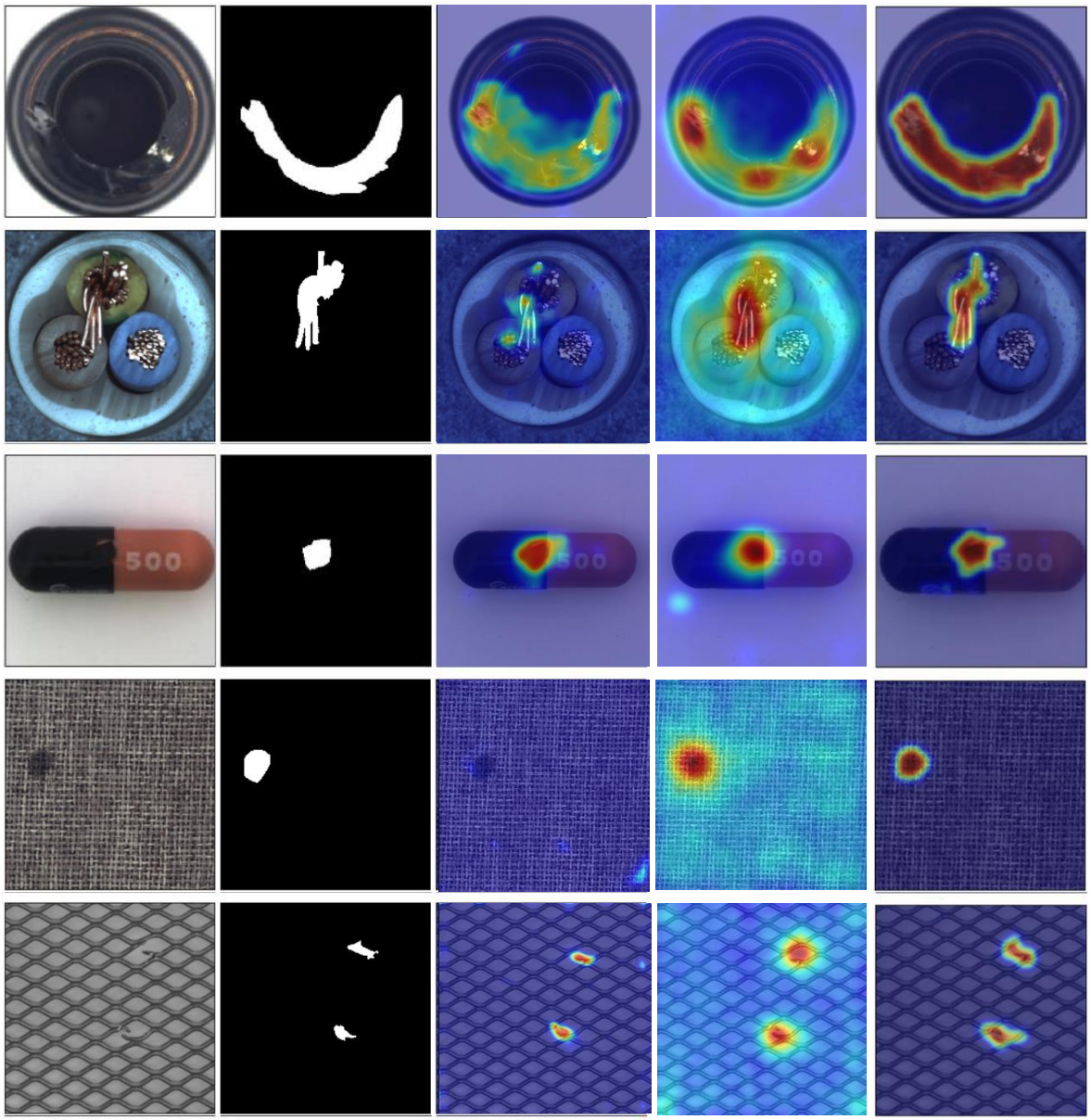


Figure 5. Visualization of anomaly detection results achieved by SOTA models on the MVTec AD dataset. Within each group, from left to right, are the anomaly image, ground-truth, and predicted anomaly scores of each model.

References

[1] Anas Al-lahham, Nurbek Tastan, Zaigham Zaheer, and Karthik Nandakumar. A coarse-to-fine pseudo-labeling (C2FPL) framework for unsupervised video anomaly detection. *arXiv preprint arXiv:2310.17650*, 2023. 1

[2] Jaehyeok Bae, Jae-Han Lee, and Seyun Kim. PNI: industrial anomaly detection using position and neighborhood information. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6373–6383, 2023. 1, 3

[3] Paul Bergmann, Michael Fauser, David Sattlegger, and Carsten Steger. MVTec AD — A Comprehensive Real-World Dataset for Unsupervised Anomaly Detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9584–9592, 2019. 7, 10, 11, 17, 18

[4] Markus M Breunig, Hans-Peter Kriegel, Raymond T Ng, and Jörg Sander. Lof: identifying density-based local outliers. In *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, pages 93–104, 2000. 3

[5] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 9650–9660, 2021. 17, 18, 19

[6] Raghavendra Chalapathy and Sanjay Chawla. Deep learning for anomaly detection: A survey. *arXiv preprint arXiv:1901.03407*, 2019. 1

[7] Yuanhong Chen, Yu Tian, Guansong Pang, and Gustavo Carneiro. Deep one-class classification via interpolated gaussian descriptor. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 383–392, 2022. 2

[8] Niv Cohen and Yedid Hoshen. Sub-image anomaly detection with deep pyramid correspondences. *arXiv preprint arXiv:2005.02357*, 2020. 1, 3

[9] Thomas Defard, Aleksandr Setkov, Angelique Loesch, and Romaric Audigier. PaDim: a patch distribution modeling framework for anomaly detection and localization. *ICPR 2020: 25th International Conference on Pattern Recognition Workshops and Challenges*, 12664:475–489, 2021. 1, 3, 10, 21, 22

[10] Hanqiu Deng and Xingyu Li. Anomaly detection via reverse distillation from one-class embedding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9737–9746, 2022. 1, 3

[11] Choubo Ding, Guansong Pang, and Chunhua Shen. Catching both gray and black swans: Open-set supervised anomaly detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7388–7398, 2022. 1

[12] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. 17

[13] Min Du, Ruoxi Jia, and Dawn Song. Robust anomaly detection and backdoor attack detection via differential privacy. In *International Conference on Learning Representations*, 2020. 10

[14] Jia-Chang Feng, Fa-Ting Hong, and Wei-Shi Zheng. MIST: Multiple instance self-training framework for video anomaly detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14009–14018, 2021. 1

[15] Aritra Ghosh, Himanshu Kumar, and P Shanti Sastry. Robust Loss Functions under Label Noise for Deep Neural Networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31, 2017. 9

[16] Yibin Huang, Congying Qiu, and Kui Yuan. Surface defect saliency of magnetic tile. *The Visual Computer*, 36:85–96, 2020. 18

[17] Jiin Im, Yongho Son, and Je Hyeong Hong. FUN-AD: Fully Unsupervised Learning for Anomaly Detection with Noisy Training Data, 2025. 17

[18] Jiin Im, Yongho Son, and Je Hyeong Hong. Supplementary document for fun-ad: Fully unsupervised learning for anomaly detection with noisy training data, 2025. 3, 7, 10, 11

[19] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009. 10, 17, 18, 19

[20] Chun-Liang Li, Kihyuk Sohn, Jinsung Yoon, and Tomas Pfister. Cutpaste: Self-supervised learning for anomaly detection and localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9664–9674, 2021. 1, 3

[21] Tangqing Li, Zheng Wang, Siying Liu, and Wen-Yan Lin. Deep unsupervised anomaly detection. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 3636–3645, 2021. 2

[22] Zhikang Liu, Yiming Zhou, Yuansheng Xu, and Zilei Wang. Simplenet: A simple network for image anomaly detection and localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 20402–20411, 2023. 1, 3, 10, 17, 21, 22

[23] Xiaoxiao Ma, Jia Wu, Shan Xue, Jian Yang, Chuan Zhou, Quan Z Sheng, Hui Xiong, and Leman Akoglu. A comprehensive survey on graph anomaly detection with deep learning. *IEEE Transactions on Knowledge and Data Engineering*, 2021. 1

[24] Larry M Manevitz and Malik Yousef. One-Class SVMs for Document Classification. *Journal of Machine Learning Research*, 2(Dec):139–154, 2001. 1

[25] Declan McIntosh and Alexandra Branzan Albu. Inter-realization channels: Unsupervised anomaly detection beyond one-class classification. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 6285–6295, 2023. 1, 2, 3, 10, 11, 18, 19, 21, 22

[26] Guansong Pang, Cheng Yan, Chunhua Shen, Anton van den Hengel, and Xiao Bai. Self-trained deep ordinal regression for end-to-end video anomaly detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12173–12182, 2020. 1

[27] Chen Qiu, Aodong Li, Marius Kloft, Maja Rudolph, and Stephan Mandt. Latent outlier exposure for anomaly detection with contaminated data. In *Proceedings of the 39th International Conference on Machine Learning*, pages 18153–18167. PMLR, 2022. 10

[28] Karsten Roth, Latha Pemula, Joaquin Zepeda, Bernhard Schölkopf, Thomas Brox, and Peter Gehler. Towards total recall in industrial anomaly detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14318–14328, 2022. 1, 3, 10, 21, 22

[29] Marco Rudolph, Tom Wehrbein, Bodo Rosenhahn, and Bastian Wandt. Fully convolutional cross-scale-flows for image-based defect detection. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 1088–1097, 2022. 1, 3, 10, 21, 22

[30] Hannah M Schlüter, Jeremy Tan, Benjamin Hou, and Bernhard Kainz. Natural synthetic anomalies for self-supervised anomaly detection and localization. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 474–489, 2022. 1, 3

[31] Ozan Sener and Silvio Savarese. Active Learning for Convolutional Neural Networks: A Core-Set Approach. In *International Conference on Learning Representations*, 2018. 18

[32] Woosang Shin, Jonghyeon Lee, Taehan Lee, Sangmoon Lee, and Jong Pil Yun. Anomaly detection using score-based perturbation resilience. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 23372–23382, 2023. 1, 3

[33] Tran Dinh Tien, Anh Tuan Nguyen, Nguyen Hoang Tran, Ta Duc Huy, Soan Duong, Chanh D Tr Nguyen, and Steven QH Truong. Revisiting reverse distillation for anomaly detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 24511–24520, 2023. 1, 3

[34] Dong Wang and Xiaoyang Tan. Unsupervised feature learning with c-svddnet. *Pattern Recognition*, 60:473–485, 2016. 18, 19

[35] Gaoang Wang, Yibing Zhan, Xinchao Wang, Mingli Song, and Klara Nahrstedt. Hierarchical semi-supervised contrastive learning for contamination-resistant anomaly detection. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 110–128, 2022. 18

[36] Shenzhi Wang, Liwei Wu, Lei Cui, and Yujun Shen. Glancing at the patch: Anomaly localization with global and local feature comparison. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 254–263, 2021. 1, 3

[37] Julia Wolleb, Florentin Bieder, Robin Sandkühler, and Philippe C Cattin. Diffusion models for medical anomaly detection. In *International Conference on Medical image computing and computer-assisted intervention*, pages 35–45, 2022. 1

[38] Jiang Xi, Jianlin Liu, Jinbao Wang, Qiang Nie, Kai WU, Yong Liu, Chengjie Wang, and Feng Zheng. SoftPatch: Unsupervised anomaly detection with noisy data. *Advances in Neural Information Processing Systems*, 35:15433–15445, 2022. 1, 2, 3, 10, 11, 17, 18, 19, 21, 22

[39] Tiange Xiang, Yixiao Zhang, Yongyi Lu, Alan L Yuille, Chaoyi Zhang, Weidong Cai, and Zongwei Zhou. Squid: Deep feature in-painting for unsupervised anomaly detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 23890–23901, 2023. 1

[40] Guoyang Xie, Jinbao Wang, Jiaqi Liu, Jiayi Lyu, Yong Liu, Chengjie Wang, Feng Zheng, and Yaochu Jin. Im-iad: Industrial image anomaly detection benchmark in manufacturing. *arXiv preprint arXiv:2301.13359*, 2023. 1

[41] Xincheng Yao, Ruoqi Li, Jing Zhang, Jun Sun, and Chongyang Zhang. Explicit boundary guided semi-push-pull contrastive learning for supervised anomaly detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 24490–24499, 2023. 1, 2

[42] Jinsung Yoon, Kihyuk Sohn, Chun-Liang Li, Sercan O Arik, Chen-Yu Lee, and Tomas Pfister. Self-supervise, Refine, Repeat: Improving Unsupervised Anomaly Detection. In *Transactions on Machine Learning Research*, pages 2835–8856, 2022. 2, 3

[43] Guang Yu, Siqi Wang, Zhiping Cai, Xinwang Liu, Chuanfu Xu, and Chengkun Wu. Deep anomaly discovery from unlabeled videos via normality advantage and self-paced refinement. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13987–13998, 2022. 1

[44] M Zaigham Zaheer, Arif Mahmood, M Haris Khan, Mattia Segu, Fisher Yu, and Seung-Ik Lee. Generative cooperative learning for unsupervised video anomaly detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14744–14754, 2022. 1

[45] Vitjan Zavrtanik, Matej Kristan, and Danijel Skocaj. Draem-a discriminatively trained reconstruction embedding for surface anomaly detection. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 8330–8339, 2021. 1, 3

[46] Vitjan Zavrtanik, Matej Kristan, and Danijel Skocaj. Dsr–a dual subspace re-projection network for surface anomaly detection. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 539–554. Springer, 2022. 1, 3

[47] Hui Zhang, Zuxuan Wu, Zheng Wang, Zhineng Chen, and Yu-Gang Jiang. Prototypical residual networks for anomaly detection and localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16281–16291, 2023. 1, 2

[48] Xuan Zhang, Shiyu Li, Xi Li, Ping Huang, Jiulong Shan, and Ting Chen. Destseg: Segmentation guided denoising student-teacher for anomaly detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3914–3923, 2023. 1, 3

[49] Ximiao Zhang, Min Xu, and Xiuzhuang Zhou. RealNet: A Feature Selection Network with Realistic Synthetic Anomaly for Anomaly Detection. *arXiv preprint arXiv:2403.05897*, 2024. 10, 11, 21, 22

[50] Dengyong Zhou, Olivier Bousquet, Thomas Lal, Jason Weston, and Bernhard Schölkopf. Learning with local and global consistency. *Advances in Neural Information Processing Systems*, 16, 2003. 7

[51] Yang Zou, Jongheon Jeong, Latha Pemula, Dongqing Zhang, and Onkar Dabeer. Spot-the-difference self-supervised pre-training for anomaly detection and segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 392–408, 2022. 1, 10, 11

Supplementary Document for FUN-AD: Fully Unsupervised Learning for Anomaly Detection with Noisy Training Data

1. Additional statistical analysis of pairwise distances between features

Empirical validation Since our statistical analysis is limited to isotropic Gaussian distributions, it is not directly applicable other distributions or real-world data. Therefore, we aim to bridge this theoretical gap with empirical analysis using real-world data. We validate these findings on both synthetic data with isotropic Gaussian distributions and real data from the *bottle* set in MVTec AD [3], utilizing normal images from the training set and anomaly images from the test set, as the training set does not contain any anomalies.

For the synthetic experiment, we sampled 1000 16-dimensional features from the normal distribution $\mathcal{N}(\mathbf{0}, \mathtt{I})$ and 1000 samples from the anomaly distribution $\mathcal{N}(\mathbf{1}.5, 2\mathtt{I})$. We then computed all pairwise feature distances, resulting in the histogram shown in Fig. 1a. For the real experiment, we extracted both image-level features and patch-level features from all 209 normal (training) images and 63 anomaly (test) images of the *bottle* sequence in MVTec AD using the pretrained DINO model from [5]. Again, we calculated all pairwise feature distances over all pairs of patch-level features and over all pairs of image-level features, yielding histograms in Figs. 1b and 1c. While the histograms have different degrees of skewness, we observe the normal pairs are consistently the most likely to yield shorter pairwise distances compared to other types of pairs.

2. Toy example of semantic anomaly detection

We used CIFAR-10 [19] to conduct a toy experiment, setting the data to a scenario where the distribution of outliers is more spread out than the distribution of normals, consistent with our assumptions. The normal class is "automobile", and the outliers consist of the remaining classes in CIFAR-10. The contamination ratio (the ratio of outliers to normals) within the training dataset is set to 10%. Unlike detecting patch-level defects, Local-Net in semantic anomaly detection outputs one anomaly score per image (because semantic anomalies are not divided into normal and abnormal regions within a single image). For further details, please refer to Sec. 4 for related results.

3. Additional framework details

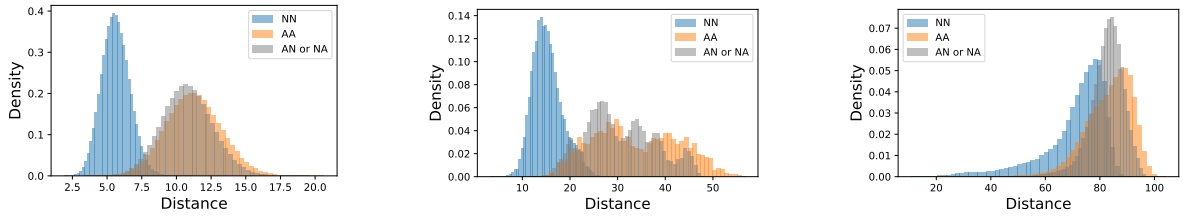
Overall architecture FUN-AD comprises two sub-networks: a pretrained feature extractor $\mathcal{E}$ (to leverage semantic information, the self-supervised DINO [5]) based on vision transformer (ViT) [12] and the Local-Net model ϕ based on a simple multilayer perceptron (MLP) for detecting patch-level anomalies. $\mathcal{E}$ takes an image I_i as input and outputs one class token and P patch tokens. To identify anomalies at the patch level, we concatenate the class token with each patch token to form a patch-level feature $\mathbf{f}_{ij} \in \mathbb{R}^D$ for each image patch X_{ij}. With abuse of notation, we represent $I_i = \{X_{ij}\}_{j=1}^P$, where $P = HW/K^2$. In our setting, $H = W = 224$, $K = 8$, and thus $P = 784$. Since the class token is 768-dimensional and the patch token is 768-dimensional, $D = 1536$. The local patch feature $\mathbf{f}_{ij}$ serves as input to the Local-Net ϕ, from which we obtain a normalized anomaly score using a sigmoid function.

Model inference In the inference phase, FUN-AD performs anomaly detection and anomaly localization by predicting the anomaly score for a given input. For anomaly detection, a test image X_i is passed through $\mathcal{E}$ to extract the patch-level features $\{\mathbf{f}_{ij}\}_{j=1}^P$. These features are passed through ϕ to obtain the patch-level anomaly scores. We then perform global max-pooling of these scores to calculate the image-level global anomaly score. For anomaly localization, the patch-level anomaly scores are spatially arranged to form an anomaly score map, as shown in Fig. 5 in [17]. As in [22, 38], we then perform bilinear interpolation of the map with Gaussian smoothing ($\sigma = 4$) to match the dimensions of the original image ($H \times W$).

Implementation details The Local-Net has the FC[1536, 1024, 128, 1] structure. Leaky ReLU activation functions (slope: 0.2) are applied between layers, and the output layer uses the sigmoid function for outputting normalized anomaly score. We use the RMSProp optimizer with momentum of 0.2 and learning rate of 2e-5 for training using the batch size of 32. We set $\lambda = 2.5$, $\tau_b = 0.5$, $\tau_n = 0.5$ and $\tau_c = 0.9$ by default. For each object/texture class, we train for 1500 epochs and choose the model with the best average of image-wise and pixel-wise AUROCs. In Sec. 3 of the main paper [17], the real data consists of normal data in the training set and anomalous data in the test set. Also, we set the patch size to 8, yielding one 768-dimensional image-level feature and 28^2 768-dimensional patch-level features for each normal or anomaly image.

4. Additional ablation studies

Effect of different contamination rates Tab. 1 demonstrates the performance of FUN-AD according to the contamination ratio in the training dataset. Here, "FUN-AD" refers to the results obtained from training with the dataset



(a) Synthetic (isotropic Gaussians) (b) Real (bottle, image features) (c) Real (bottle, patch features)

Figure 1. Histogram of pairwise distances for different types of feature pairs. Abbreviations are as follows: NN for normal-normal pairs, AA for anomaly-anomaly pairs, AN or NA for anomaly-normal or normal-anomaly pairs. For the synthetic experiment, the normal samples were drawn from $\mathcal{N}(\mathbf{0}, \mathtt{I})$ and the anomaly samples from $\mathcal{N}(\mathbf{1.5}, 2\mathtt{I})$. For the "real" comparison, we used patch features (patch tokens) and image features (class tokens) extracted from the pretrained DINO model [5] for the bottle set [3].

Dataset	Method	Anomaly-to-normal ratio					
		0%	1%	3%	5%	10%	20%
MTD [16]	InReach [25]	83.55 / 72.02	84.08 / 75.49	80.30 / 78.79	80.93 / 78.64	80.73 / 72.23	88.42 / 81.91
	SoftPatch [38]	76.11 / 90.63	77.19 / 91.53	79.61 / 92.67	80.51 / 91.48	83.61 / 87.52	85.26 / 93.49
	FUN-AD	79.55 / 94.75	82.51 / **94.58**	85.87 / 94.97	85.35 / 93.84	85.12 / 93.52	94.74 / 95.37
	FUN-AD*	**83.61 / 93.51**	**85.25** / 93.59	**87.52 / 95.22**	**85.46 / 94.13**	**85.28 / 93.59**	**95.79 / 97.76**

Table 1. Performance comparison of different fully-unsupervised anomaly detection methods across different anomaly-to-normal ratios on the contaminated MTD dataset (*no overlap*). * indicates synthetic anomaly data has been utilized for training. The best results are in bold and the runner-ups are underlined.

without synthetic anomalies, while "FUN-AD*" refers to the results from training the FUN-AD framework with synthetic anomalies added at a rate of 5% of the training dataset size. Synthetic anomalies were created from a noisy (anomaly-present) dataset considering a fully unsupervised setting. The results demonstrate that our proposed framework achieves state-of-the-art performance on texture-based dataset, highlighting its robustness across various types of anomalies.

Inference time In an industrial setting, real-time anomaly detection is crucial. When comparing the inference speed with existing methods using the GPU RTX-4090 (refer to Tab. 2), our method operates at an impressive speed of approximately 113 fps, outperforming other methods.

Effect of weight on mutual smoothness loss Tab. 3 shows that optimal performance is achieved when $\lambda = 2.5$ on MVTec AD. In these results, $\lambda = 0$ indicates that the pseudo-labeling method alone is sufficient for the network to learn from the normal and anomaly information and succeed in anomaly detection and localization. Additionally, when mutual-smoothness loss is applied, the anomaly detection performance improves with a weight value of $\lambda = 2.5$ compared to using only pseudo-labeling.

Effects of random sampling rate Since Eq.7 needs to be calculated for pseudo-labeling, the training time overhead can be significant if computed with all feature vectors in the memory bank. However, applying corset sampling [31], which has been used in a one-class classification environment, is difficult because we cannot assume that all the samples in the memory bank are normal. Therefore, we compare the performance by randomly sampling only a small percentage of the feature vectors in the memory bank. Table 4 shows the performance of anomaly detection and localization according to the sampling ratio. The performance does not vary significantly depending on the degree of sampling. This indicates that using a low sampling rate for efficient training does not result in significant performance degradation.

Effects of synthetic supervised loss The comparison with and without y_{syn} in Tab. 5 shows that our proposed pseudo labels perform better than those using Perlin masks to assign labels for synthetic anomalies. This indicates that our pseudo-labeling method is more effective for detecting real anomalies by identifying semantically anomalous regions and using them for training, rather than merely learning that regions with the Perlin noise are anomalous.

Semantic anomaly detection Detecting semantic anomalies also requires a fully unsupervised setting, and according to [35], it is more similar to the real world when the training data is contaminated with abnormal samples. Therefore, we conducted experiments on the STR-10 [34] and CIFAR-10 [19] datasets to verify the applicability of our framework. We designated one class as the normal class and randomly sampled anomalies from the remaining classes to create contaminated unlabeled datasets with a 1:10 anomaly-to-normal ratio. The findings are presented in Tab. 6, demonstrating that although FUN-AD was originally developed for industrial anomaly detection, it effectively distinguishes between normal and abnormal classes under specific conditions. In these experiments, where the normal class is singular and the abnormal classes encompass the remaining nine, the variation is substantial enough to validate the

Method	FUN-AD (*Ours*)	SoftPatch [38]	InReaCh [25]
Throughput (fps)	**112.7**	22.5	8.8

Table 2. Inference speeds achieved by different fully unsupervised anomaly detection algorithms on the VisA dataset.

λ	AUROC$_{image}$(%)	AUROC$_{pixel}$(%)
0	98.72	98.39
0.25	98.54	98.41
1	98.83	98.45
2.5	**98.95**	**98.55**
10	98.55	98.34

Table 3. Ablation study of weights for the mutual smoothness loss. The optimal performance is achieved when $\lambda = 2.5$ on MVTec AD.

Sampling ratio	AUROC$_{image}$(%)	AUROC$_{pixel}$(%)
0.25	98.83	**98.66**
0.5	98.95	98.55
0.75	**99.11**	98.51
1.0	98.84	98.53

Table 4. Ablation study of the sampling rate when storing normally labeled feature vectors in memory banks. This demonstrates the capability of efficient training with a low sampling rate.

Method	AUROC$_{image}$(%)	AUROC$_{pixel}$(%)
w/o y_{syn}	97.83	**97.51**
w y_{syn}	98.95	98.55

Table 5. Ablation study of synthetic supervised loss. y_{syn} indicates whether the mask of the synthetic anomaly is used or not.

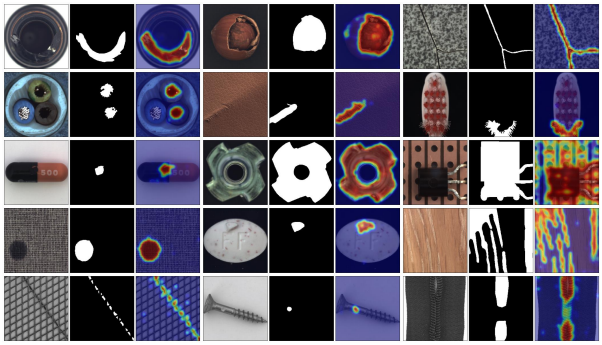
Dataset	AUROC (%)
CIFAR-10 [19]	95.10
STL-10 [34]	99.63

Table 6. Average semantic anomaly detection results for scenarios (10, e.g., cat is normal) where each semantic class is normal.

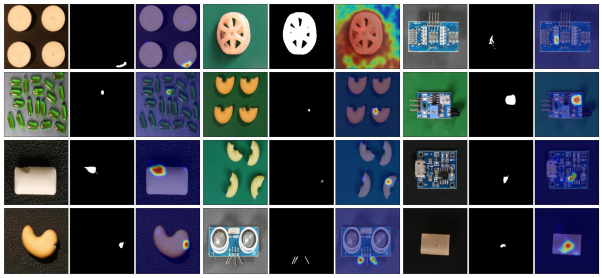
effectiveness of our assumptions and approach.

5. Qualitative results

Fig. 2 shows some anomaly localization results yielded by FUN-AD. Each class is represented by three columns: the first column shows the RGB image, the second column shows the segmentation mask of the defect area, and the third column shows the anomaly score predicted by FUN-AD. Our method is not only effective at detecting large defects but also excels at clearly separating the boundary between normal and anomaly without ambiguity, even in the presence of very small defects. This is evident when comparing the ground-truth mask and the heatmap in Fig. 2. These results demonstrate that FUN-AD is robust, particularly for very small defects, but not limited to large detects with higher confidence score compared to other models.



(a) MVTec AD



(b) VisA

Figure 2. Visualization of anomaly detection results achieved by FUN-AD on the MVTec AD and VisA datasets. Each binary mask shows the anomaly segmentation map while each heatmap visualizes the anomaly region (red means likely to be an anomaly while blue means unlikely).

6. Details of the experimental results

We show the experimental results for all categories of *overlap*, *No overlap* for MVTec AD and VisA in Tab. 7, 8, 9, 10. Each table presents image-wise AUROC (%) / pixel-wise AUROC (%), representing anomaly detection and localization performance, respectively. The best results are in bold and the runner-ups are underlined. *FUN-AD* places a stronger emphasis on local anomalies by utilizing Local-Net for inference. Consequently, it excels at detecting small defects in images with multiple instances, as observed in capsules and macaroni2 in VisA, outperforming other models in this regard.

7. Limitations and broader impacts

Limitations While FUN-AD is shown to work across many different unsupervised settings, it may be compromised if the feature diversity of the normal data is comparable to that of the anomalies, e.g. when one type of anomaly dominates. Also, our analytic analysis in Sec.3.1 is limited to the case of normal and anomaly distributions following isotropic Gaussians. Our approach still requires use of a pretrained feature extraction network such as DINO [5] for basic initialization. Finally, FUN-AD yields suboptimal performance for scarce anomaly-to-normal ratios (0 to 1%).

Broader impacts Our approach can reduce the physical burden of human workers by reducing the manual labor required for annotating normal samples. This allows reducing expenditure on data acquisition which in return may be invested towards improving the quality of product.

Type	One-class classification					Fully unsupervised		
Method	CS-Flow [29]	PaDiM [9]	PatchCore [28]	SimpleNet [22]	RealNet [49]	SoftPatch [38]	InReaCh [25]	FUN-AD (*Ours*)
Bottle	98.7 / -	99.3 / 98.4	**100.0** / 98.2	**100.0** / 97.2	**100.0** / 98.2	**100.0** / 98.7	**100.0** / 98.3	**100.0 / 99.2**
Cable	95.5 / -	89.3 / 93.6	96.9 / 80.9	97.0 / 93.7	96.1 / 95.6	**99.1 / 98.7**	94.2 / 97.5	98.7 / 94.0
Capsule	95.0 / -	90.5 / 98.5	97.2 / 98.7	95.3 / 98.1	**99.8** / 98.6	95.8 / **98.9**	49.7 / 93.4	99.7 / 98.2
Carpet	99.8 / -	**100.0** / 99.2	98.7 / 99.1	99.4 / 98.8	98.6 / 97.7	99.1 / 99.4	98.4 / 99.5	**100.0 / 99.7**
Grid	96.5 / -	93.4 / 95.6	96.2 / 98.9	99.1 / 97.9	**100.0 / 99.6**	96.3 / 98.7	91.2 / 97.7	**100.0** / 99.4
Hazelnut	95.1 / -	92.7 / 98.0	**100.0** / 98.8	97.6 / 95.6	99.9 / 98.5	**100.0** / 98.8	98.3 / 97.8	**100.0 / 99.7**
Leather	98.6 / -	**100.0** / 99.3	**100.0** / 99.2	**100.0** / 98.9	**100.0** / 99.7	**100.0** / 99.4	**100.0** / 99.3	**100.0 / 99.8**
Metal_nut	91.8 / -	97.7 / 89.3	99.1 / 77.6	99.0 / 85.8	**100.0** / 87.0	**100.0** / 86.8	96.8 / 95.1	**100.0 / 99.5**
Pill	89.3 / -	93.7 / 96.2	97.0 / 97.0	95.6 / 97.9	**99.1 / 98.9**	96.7 / 97.8	88.6 / 96.0	98.5 / 98.2
Screw	79.3 / -	84.5 / 98.4	95.0 / 98.6	89.0 / 97.8	**98.8 / 99.4**	94.5 / **99.4**	79.7 / 98.3	94.8 / 98.1
Tile	96.2 / -	97.8 / 94.9	99.2 / 92.8	99.4 / 91.6	**99.7** / 97.8	98.7 / 96.3	99.0 / 97.3	99.5 / **98.7**
Toothbrush	92.7 / -	**100.0** / 98.8	99.7 / 98.8	**100.0** / 98.4	98.3 / 95.0	99.7 / 98.6	99.0 / **98.8**	97.9 / 98.7
Transistor	92.3 / -	94.6 / 96.8	96.8 / 87.7	96.0 / 89.7	98.0 / 90.9	**99.6** / 93.6	99.2 / 97.1	99.3 / **97.7**
Wood	90.5 / -	98.1 / 94.3	96.6 / 95.6	99.7 / 92.9	99.9 / 97.3	98.1 / 95.1	95.3 / 92.3	**100.0 / 98.0**
Zipper	95.1 / -	88.2 / 98.4	98.7 / 98.1	98.7 / 95.2	99.6 / 98.9	97.5 / 98.9	93.2 / 94.9	**100.0 / 99.3**
Average	93.8 / -	94.7 / 96.6	98.1 / 94.7	97.7 / 95.3	**99.2** / 96.9	98.3 / 97.3	92.2 / 96.9	**99.2 / 98.6**

Table 7. Detailed results for MVTec AD in the *No overlap* setting.

Type	One-class classification					Fully unsupervised		
Method	CS-Flow [29]	PaDiM [9]	PatchCore [28]	SimpleNet [22]	RealNet [49]	SoftPatch [38]	InReaCh [25]	FUN-AD (*Ours*)
Bottle	67.4 / -	79.9 / 96.0	89.4 / 68.2	76.2 / 46.3	99.3 / 97.7	**100.0** / 93.1	**100.0** / 98.4	**100.0 / 99.2**
Cable	72.7 / -	68.0 / 86.0	87.1 / 65.0	41.2 / 51.7	88.3 / 93.5	**99.3 / 98.4**	94.6 / 97.8	96.9 / 95.4
Capsule	77.7 / -	81.4 / 98.3	88.8 / 92.3	43.0 / 61.0	**97.8 / 99.2**	95.3 / 98.7	52.1 / 94.3	97.5 / 98.1
Carpet	68.4 / -	89.0 / 97.1	75.4 / 64.6	67.6 / 71.1	98.9 / 98.2	99.7 / 98.5	98.7 / 99.4	**100.0 / 99.7**
Grid	52.5 / -	52.0 / 85.7	61.3 / 60.3	57.6 / 45.4	**100.0** / 99.1	97.1 / 97.6	92.4 / 97.6	99.8 / **99.2**
Hazelnut	42.1 / -	43.2 / 95.8	67.0 / 59.7	69.4 / 64.1	99.6 / 98.5	**100.0** / 95.4	98.7 / 97.6	**100.0 / 99.7**
Leather	72.9 / -	94.7 / 97.2	81.0 / 66.9	45.8 / 31.5	**100.0** / 99.4	**100.0**/ 99.3	**100.0** / 99.0	**100.0 / 99.8**
Metal_nut	70.1 / -	74.6 / 79.9	90.2 / 62.7	60.9 / 60.7	97.2 / 86.3	99.7 / 77.0	97.2 / 98.4	**99.8 / 99.4**
Pill	72.8 / -	76.5 / 95.9	84.5 / 90.8	49.1 / 55.6	96.5 / **98.5**	94.6 / 97.3	89.1 / 96.1	**98.1 / 98.5**
Screw	58.0 / -	62.5 / 96.7	78.0 / 77.2	53.6 / 61.9	91.2 / **98.9**	**93.5** / 94.5	79.9 / 98.3	93.4 / 98.4
Tile	69.8 / -	71.0 / 72.6	86.8 / 63.3	66.9 / 33.0	98.6 / 94.2	**99.4** / 94.0	99.1 / 97.7	99.0 / **98.8**
Toothbrush	83.9 / -	80.0 / 95.3	95.3 / 91.1	50.6 / 34.9	99.2 / 94.1	**99.7 / 98.8**	98.3 / **98.8**	98.6 / **98.8**
Transistor	43.8 / -	45.8 / 90.8	72.1 / 61.4	61.5 / 48.3	88.5 / 87.6	**99.1** / 90.4	99.0 / 95.2	98.7 / **97.7**
Wood	54.3 / -	66.8 / 90.2	76.0 / 62.0	79.2 / 64.6	97.6 / 96.1	98.9 / 95.2	95.2 / 91.8	**100.0 / 98.0**
Zipper	75.9 / -	77.3 / 96.9	90.7 / 76.4	66.9 / 63.4	99.7 / 98.5	97.6 / 98.7	91.6 / 94.4	**99.9 / 99.2**
Average	65.5 / -	70.8 / 91.6	81.6 / 70.8	59.3 / 52.9	96.8 / 96.0	98.3 / 95.1	92.4 / 97.2	**98.8 / 98.6**

Table 8. Detailed results for MVTec AD in the *Overlap* setting.

Type	One-class classification					Fully unsupervised		
Method	CS-Flow [29]	PaDiM [9]	PatchCore [28]	SimpleNet [22]	RealNet [49]	SoftPatch [38]	InReaCh [25]	FUN-AD (*Ours*)
Candle	86.7 / -	91.2 / 99.4	95.0 / 99.2	93.9 / 97.9	**94.7 / 99.6**	93.8 / **99.6**	90.1 / 98.7	94.4 / 99.5
Capsules	68.2 / -	56.6 / 94.0	69.5 / 96.2	76.5 / 96.8	82.5 / 98.9	69.2 / 97.7	57.9 / 93.0	**93.8 / 99.5**
Cashew	86.1 / -	90.0 / 99.0	94.7 / 99.0	90.7 / 99.5	84.3 / 98.2	95.2 / 99.1	77.1 / 98.2	**96.6 / 99.7**
Chewinggum	93.8 / -	95.2 / 98.9	98.4 / 98.5	96.0 / 97.5	**99.8 / 99.9**	98.7 / 99.1	77.8 / 98.2	99.3 / 99.7
Fryum	78.0 / -	89.1 / 97.7	89.4 / 91.2	91.5 / 94.8	89.1 / 94.5	92.0 / 96.0	86.3 / 96.4	**96.9 / 98.2**
Macaroni1	81.4 / -	83.2 / 99.1	86.5 / 97.3	88.7 / 98.1	**98.1 / 99.9**	89.8 / 98.8	83.7 / 98.0	96.4 / **99.9**
Macaroni2	60.6 / -	60.4 / 95.6	64.8 / 89.3	72.1 / 94.6	**90.0 / 99.6**	57.6 / 95.1	57.4 / 95.8	87.0 / 99.1
PCB1	90.7 / -	94.2 / 99.6	93.0 / 86.5	93.0 / 94.8	93.8 / 99.4	95.1 / **99.8**	93.8 / 99.6	**96.2** / 99.6
PCB2	85.8 / -	91.8 / 99.2	**96.3** / 98.8	94.8 / 99.0	95.5 / 96.9	93.9 / **99.3**	91.9 / 98.7	91.2 / 97.8
PCB3	84.3 / -	85.7 / 99.1	93.8 / 96.9	95.2 / 99.2	**95.9** / 99.1	92.3 / **99.4**	93.3 / 99.3	91.4 / 98.6
PCB4	95.3 / -	97.1 / 98.2	98.0 / 96.3	97.8 / 96.1	98.9 / 98.9	99.2 / 99.1	**99.7 / 99.5**	97.6 / **99.5**
Pipe_fryum	77.3 / -	95.2 / 98.3	98.5 / 99.2	94.6 / 99.7	98.8 / 99.3	98.8 / 99.5	96.7 / **99.8**	**99.3 / 99.8**
Average	82.3 / -	85.8 / 98.3	89.8 / 95.7	90.4 / 96.7	93.5 / 98.7	89.6 / 98.5	83.8 / 97.6	**95.0 / 99.2**

Table 9. Detailed results for VisA in the *No overlap* setting.

Type	One-class classification					Fully unsupervised		
Method	CS-Flow [29]	PaDiM [9]	PatchCore [28]	SimpleNet [22]	RealNet [49]	SoftPatch [38]	InReaCh [25]	FUN-AD (*Ours*)
Candle	68.1 / -	79.7 / 99.1	85.3 / 87.5	47.6 / 49.0	**95.3 / 99.4**	93.7 / **99.4**	85.2 / 95.5	93.8 / **99.4**
Capsules	48.6 / -	45.3 / 78.9	65.2 / 83.9	50.3 / 50.2	82.6 / 96.8	70.5 / 89.2	53.1 / 89.0	**93.3 / 99.4**
Cashew	64.4 / -	72.0 / 85.2	86.8 / 85.7	61.4 / 60.8	88.9 / 96.6	94.1 / 98.9	75.5 / 91.0	**96.9 / 99.7**
Chewinggum	57.8 / -	76.2 / 82.1	87.9 / 78.1	51.9 / 59.9	**99.9** / 99.4	97.9 / 98.9	76.1 / 90.4	99.2 / **99.7**
Fryum	73.1 / -	71.3 / 92.9	84.3 / 82.5	51.9 / 80.0	86.1 / 94.8	92.9 / 92.4	78.1 / 94.7	**96.5 / 98.2**
Macaroni1	65.1 / -	68.8 / 97.5	80.2 / 84.0	45.7 / 49.3	**98.1 / 99.8**	89.0 / 97.2	75.5 / 88.1	96.7 / **99.8**
Macaroni2	48.2 / -	48.4 / 89.7	58.2 / 80.4	51.2 / 57.0	**88.8 / 99.2**	56.5 / 85.9	51.0 / 91.4	87.7 / **99.2**
PCB1	72.5 / -	77.4 / 98.0	87.3 / 71.4	44.7 / 60.8	93.2 / 98.9	95.9 / **99.6**	94.4 / 97.2	**96.7 / 99.6**
PCB2	68.7 / -	74.1 / 96.4	86.3 / 83.5	43.9 / 46.8	91.0 / 96.5	**93.0 / 98.0**	86.7 / 94.1	91.5 / **98.0**
PCB3	67.5 / -	69.6 / 97.0	85.6 / 76.6	53.7 / 65.8	90.0 / 96.3	**92.4 / 98.7**	83.1 / 97.3	91.7 / 98.2
PCB4	76.3 / -	82.1 / 95.7	93.9 / 83.0	46.9 / 68.0	97.0 / 96.6	**99.2** / 97.9	99.0 / 96.7	97.8 / **99.5**
Pipe_fryum	61.8 / -	76.2 / 95.7	87.1 / 82.6	56.2 / 47.2	98.1 / 97.7	98.4 / 99.4	84.4 / 97.5	**99.4 / 99.8**
Average	64.3 / -	70.1 / 92.3	82.3 / 81.6	50.4 / 57.9	92.4 / 97.7	89.5 / 96.3	78.5 / 93.6	**95.1 / 99.2**

Table 10. Detailed results for VisA in the *Overlap* setting.