arXiv:2504.02775v2 [cs.CV] 26 May 2026

TailedCore: Few-Shot Sampling for Unsupervised Long-Tail Noisy Anomaly Detection

Yoon Gyo Jung[*1] Jaewoo Park[*‡2] Jaeho Yoon[*3] Kuan-Chuan Peng[4]
Wonchul Kim[2] Andrew Beng Jin Teoh[3] Octavia Camps[†1]

[1]Northeastern University [2]AiV Co. [3]Yonsei University

[4]Mitsubishi Electric Research Laboratories

Abstract

We aim to solve unsupervised anomaly detection in a practical challenging environment where the normal dataset is both contaminated with defective regions and its product class distribution is tailed but unknown. We observe that existing models suffer from tail-versus-noise trade-off where if a model is robust against pixel noise, then its performance deteriorates on tail class samples, and vice versa. To mitigate the issue, we handle the tail class and noise samples independently. To this end, we propose TailSampler, a novel class size predictor that estimates the class cardinality of samples based on a symmetric assumption on the class-wise distribution of embedding similarities. TailSampler can be utilized to sample the tail class samples exclusively, allowing to handle them separately. Based on these facets, we build a memory-based anomaly detection model TailedCore, whose memory both well captures tail class information and is noise-robust. We extensively validate the effectiveness of TailedCore on the unsupervised long-tail noisy anomaly detection setting, and show that TailedCore outperforms the state-of-the-art in most settings. Code is available in TailedCore.

1. Introduction

In the complex landscape of anomaly detection, the challenge often lies in navigating practical scenarios that feature a diverse and imbalanced data distribution [6, 8, 11, 21]. We delve into a challenging yet highly realistic scenario in anomaly detection, where the training dataset, comprising multiple product classes [39], is beset by two significant complications: *contamination with noise* [20] and *presence of tailed class distributions* [25]. In this context, 'tail' (few-shot) classes have limited few-shot samples, unlike 'head' (many-shot) classes which are data-rich but might contain samples with defects. Crucially, in this unsupervised setting, the class information of these products remains unknown to the model trainer. This setting mirrors a common predicament in large-scale industrial applications, where companies are required to manage vast production lines encompassing a wide array of product types, each with varying production rates.

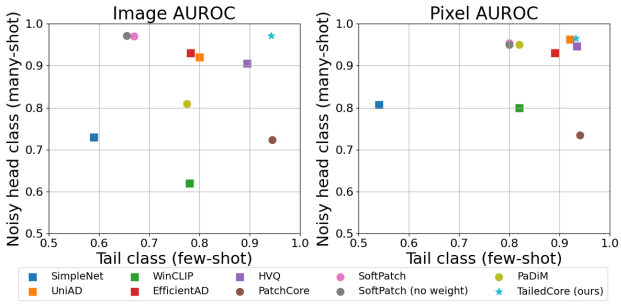


Figure 1. Tail class (x-axis) and noisy head class (y-axis) performance comparison. The tail-versus-noise trade-off is shown across memory-based anomaly detection models (circles), and is more indicative in anomaly classification task evaluated by image-level AUROC (left).

A critical issue arises in this scenario, which we term the *"tail-versus-noise dilemma."* The dilemma underscores a fundamental trade-off where *models robust to class imbalance overfit to defect noises*, likely misclassifying many defects as normal. In contrast, existing *models that are robust against defect noise tend to underfit*, missing vital details in tail classes. Fig. 1 shows this tail-versus-noise trade-off which is prevalent through existing anomaly detection algorithms. This intrinsic characteristic comes from the fact that few-shot class features are equivalent to the noisy abnormal features in terms of their statistical occurrence. Consequently, existing models find it difficult to strike a balance between avoiding overfitting to noise and effectively capturing the nuances of less represented classes.

To resolve this challenge, we propose to handle tail classes independently from head classes by exclusively sampling them. To this end, we devise a novel class size predictor TailSampler that estimates the cardinality of the class of any training sample. For accurate estimation, TailSampler estimates the class sizes based on a reflective symmetry between inter-class

[*]Equal contribution. [‡]Project lead. [†]Corresponding author.

and intra-class similarity distributions in the embedding space. The estimated class sizes of the training samples then enable to sample tail class features exclusively from the training set.

By utilizing the aggregated few-shot class instances, we devise a memory-based anomaly detector TailedCore, which is built upon the noise-discriminated model by augmenting the tail class patch features sampled with TailSampler. As the memory of TailedCore is both clean from noisy defect patches and preserves the few-shot class information, it exhibits robustness against both noise contamination and class imbalance of the training set.

Our contributions are summarized as follows:

- We explore a practical and challenging anomaly detection scenario, where a single detection model is trained on multi-class product training samples that involve noise contamination and whose class distribution is long-tailed.
- We propose a memory-based anomaly detector TailedCore whose patch memory bank is both noise-free and augmented with the representative patches of few-shot class instances. To build TailedCore, we introduce TailSampler that can aggregate few-shot class instances exclusively. TailSampler is based on a novel class size predictor that estimates the class size of training samples based on the reflective symmetry between inter-class and intra-class class-wise embedding similarity distributions.
- We conduct extensive analyses and comparative studies on both TailedCore and few-shot sampler along with the proposed class size predictor on the proposed unsupervised long-tail noisy anomaly detection benchmark.

The codes for TailedCore along with the data generation will be open sourced. We use the term 'few-shot' and 'tail' interchangeably to denote the class with a small number of samples.

2. Related works

2.1. Anomaly detection in extreme scenarios

One vitally important aspect of anomaly detection [9, 34] is the environment where these models are tested. This exploration has led to the identification of several challenging scenarios, including multi-class settings with training data spanning multiple product types [26, 39], zero-shot and few-shot settings with minimal training images [15, 19, 23, 38, 43], and noisy settings featuring uncleaned training data with defective images [7, 20, 28].

Multi-class anomaly detection In contrast to single-product settings, anomaly detection models trained on multi-class datasets often encounter "shortcut learning" [39], leading to low anomaly scores for normal images. This issue is particularly pronounced in reconstruction-based models, which tend to reconstruct all samples, regardless of their normality or abnormality. To resolve it, UniAD [39] employs neighbor masked attention within a transformer model, focusing attention on local parts to prevent shortcut learning. HVQ [26] utilizes vector quantization to discretize the latent space, restricting the network's reconstruction fidelity on normal data distributions. UniFormaly [22] increases sensitivity to irrelevant patterns by masking less relevant patches and employing top-k matchings with a pretrained foundation model. These multi-class anomaly detection methods do not consider either long-tail classes or contaminated training data setups, and do not work well in the combined "noisy long-tail" setup, which is shown in our experimental results.

Zero/few-shot anomaly detection The inception of few-shot learning benchmarks in anomaly detection marked a significant advancement, with methods leveraging template matching [18] and graph representations [38] for minimal training data scenarios. AnomalyCLIP [43], WinCLIP [19], and AnomalyGPT [17], on the other hand, detect anomalies in the few-shot classes based on the acquired knowledge of pretrained foundation models such as CLIP [33] and GPT [32]. However, these methods often require sophisticated prompt engineering or delicate prompt-tuning, and have yet to achieve optimal precision and recall in the single-class training setting with a sufficient amount of training samples. In contrast, our proposed TailedCore needs none of these process.

Noise anomaly detection Dealing with training datasets contaminated with defective samples presents another extreme learning scenario. SoftPatch [20] and InReaCh [28] focus on purifying memory coresets through outlier detection algorithms and distance association filtering, respectively. IGD [7] adopts a robust Gaussian modeling approach, addressing both contaminated training data and small dataset challenges. Yet, these methods do not fully consider scenarios involving both data contamination and class imbalance, unlike our proposed TailedCore.

2.2. Long-tail noisy learning

In the realm of image classification, several studies have focused on learning scenarios involving training sets with label contamination and class imbalance. To address these challenges, the existing literature relies on the robustness and unsupervised aspects of self-supervised learning [14, 27, 37, 41]. The learned embeddings of self-supervised frameworks are less impacted by label contamination and class imbalances. However, these works differ from ours in that they deal with label-level contamination rather than pixel-level defects. Furthermore, these methods require explicit supervised labels to remediate the contamination. In contrast, our work targets noise in pixels without any class label information.

2.3. Few-shot/outlier sampling

Both classical clustering methods and outlier detection algorithms can be applied to our task, as they facilitate the identification of few-shot tail class samples, either indirectly or directly. Unsupervised clustering algorithms [2, 13, 16] can estimate the cluster sizes but struggle to identify small sample clusters in the imbalanced data. DBSCAN [13], on the other hand, captures both clusters and outliers. This however can be problematic when attempting to identify clusters with only one

sample (1-shot) as it may be confused between few-shot samples and outliers. Unlike clustering methods, outlier detection algorithms [1, 5, 24, 35] are adept at identifying 1-shot samples which inherently resemble outliers. However, their effectiveness diminishes when a few-shot class contains more than one sample, as these no longer fit the typical outlier profile, potentially leading to incorrect identification of few-shot class samples. Both algorithm types, due to their inherent design, struggle to accurately predict class/cluster sizes in datasets that are both imbalanced and contaminated. Our proposed class size predictor specifically addresses this issue, aiming to reliably and accurately identify few-shot class samples and enable their exclusive sampling. Ultimately, the identified few-shot class samples are augmented to the noise-discriminated patch set, constituting the memory-based anomaly detection model TailedCore, which is robust against both class imbalance and training set noise.

3. Background

3.1. Memory-based anomaly detection

To train a memory-based anomaly detector, a feature extractor f (*e.g.*, WideResNet) extracts a feature map $\phi_i \in \mathbb{R}^{C\times H\times W}$ from each training input image x_i. Then, a coreset selection algorithm $\mathcal{S}_{core}$ is applied to the set $P = \{\phi_i^{(h,w)}\}$ of patch features $\phi_i^{(h,w)}$, resulting in a memory bank

$$M_{PatchCore} := \mathcal{S}_{core}(P) \subset P \tag{1}$$

that contains representative patch features of P. For instance, PatchCore opts for greedy sampling strategy to obtain the coreset. In the inference stage, the anomaly score $s^{(h,w)}$ of the testing image x is computed for each patch feature $\phi^{(h,w)}$ by measuring its distance to the memory

$$score^{(h,w)} = dist(\phi^{(h,w)}, M), \tag{2}$$

where $dist$ is the nearest neighbor distance in [34]. The pixel-level anomaly score is computed by up-scaling the score map $score = [score^{(h,w)}] \in \mathbb{R}^{H,W}$, and its image-level anomaly score is defined as its pixel maximum $\max_{h,w} score^{(h,w)}$.

3.2. Noise discrimination of memory

The training set $X = \{x_i\}$ may be contaminated with defect samples. The naively attained memory bank then contains abnormal patch features, and assigns a low anomaly score on defect images, resulting in poor performance. To resolve this, [20] applies a noise discrimination algorithm $\mathcal{S}_{clean}$ on the patch feature set P prior to coreset sampling:

$$\mathcal{S}_{clean}(P) = \{\phi_i^{(h,w)} : outlier(\phi_i^{(h,w)}) < \tau_{0.15}\}, \tag{3}$$

aggregating only the clean normal patches with low outlier scores. SoftPatch [20] exploits local outlier factor (LOF) for outlier scoring, and selects the threshold $\tau_{0.15}$ to remove only 15% of whole patches. In inference, SoftPatch utilizes the memory set $M_{SoftPatch} = \mathcal{S}_{core}(\mathcal{S}_{clean}(P))$

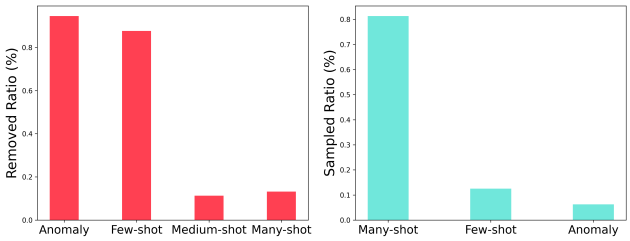


Figure 2. (left) The ratio of removed patches based on highest outlier scores by Eq. (3), which shows that most of few-shot class patches are lost. (right) The ratio of sampled patches by greedy coreset sampling from PatchCore which favors both few-shot and anomaly samples.

4. Problem and motivation

4.1. Problem

We consider a practical anomaly detection setting where the normal training data $X = \{x_i\}$ is both multi-class with tail distribution and noise-contaminated. Particularly, each x_i is paired with the product class label $y_i \in \mathcal{Y} = \{1, 2, \dots, |\mathcal{Y}|\}$ (*e.g.*, hazelnut, screw, pill, *etc.*) and an anomaly label $y_i^a \in \mathcal{Y}^a := \{0, 1\}$ with 1 indicating anomaly. However, this labeling information is ***unknown*** to the model trainer. The class distribution is long-tailed such that there are tail (*i.e.*, few-shot) classes whose number of training samples is very low; *i.e.*,

$$|C_k| = |\{x_i : y_i^c = k\}| \le K \tag{4}$$

with a very small K (*e.g.*, 1, 4, or 20), where $|\cdot|$ indicates the number of elements in the set. Moreover, some of the training samples in X are anomalies, containing defect blobs in their images. We assume the tail class samples are all normal since in the unsupervised setting its normality is determined by its majority. Overall, the training set X with the above descriptions constitutes *unsupervised tailed noisy anomaly detection* setting.

4.2. Motivation: Tail-versus-noise trade-off

An ideal model to handle a long-tail noisy dataset should selectively sample the few-shot samples while removing noisy samples. However, we observe that the existing models suffer from a trade-off between class-imbalance and noise robustness. For memory-based noise-discriminative models in particular (*e.g.*, SoftPatch), the methods focus on sampling the majority patch features, which often ignores less dominating but important patches.

As shown in Fig. 2 left, the noise discrimination process removes both defect and few-shot class patches, losing most of the information in the tail distribution. On the other hand, PatchCore samples few-shot samples well due to greedy coreset sampling which captures maximally different features. This sampling principle, however, favors the noise patch features as well as shown in Fig. 2 (right) and the low performance on noisy head classes in Fig. 1. Previous methods, hence, face the tail-versus-noise dilemma described above.

To resolve this, we propose to handle the tail classes independently from the head classes that potentially contain defective local regions. This, however, needs exclusive sampling of tail classes. To achieve this, we use the globally average pooled embedding vector, which is invariant against local pixel variation and rather exclusively corresponds to the object-centric nature of image, *i.e.*, class.

5. Method

We aim to build a coreset-based anomaly detection model using patch features. We require its memory to aptly capture tail class information and also to be absent of noise patches as much as possible. Particularly, we augment the noise discriminated memory M_{clean} with the tail class memory M_{tail}, which contains the patches of few-shot class samples exclusively:

$$M_{TailedCore} = M_{clean} \cup M_{tail}. \tag{5}$$

The main challenge lies on sampling the tail class patches exclusively. To this end, we devise a few-shot sampler by estimating the class size of samples based on its embedding (*e.g.*, global average pooling layer of the encoder). Using the estimated class sizes of training samples, the few-shot sampler enables us to exclusively sample tail class samples and the patches thereof.

5.1. Few-shot sampler: TailSampler

TailSampler first estimates the class size of every training sample. Then, by automatically determining the maximum number of tail classes, denoted by $y_{\max}$, TailSampler collects the tail class samples' patch features exclusively.

5.1.1. Class size predictor

To sample patches exclusively from few-shot class, we identify samples from few-shot classes by estimating their *"class sizes"*, namely the size of class each sample belongs to. To accurately measure the class size, we hypothesize as follows: In the embedding space, samples within an appropriate angle are majorly of the same class, and this angle is likely to be the half of maximum angle to another embedding. In addition, the neighborhoods of neighborhoods within the half angle give accurate class size estimate. Based on these hypotheses, we proceed as follows:

First, we adaptively set angle α_i for each embedding e_i from training samples x_i. The coverage of angle α_i decides the neighbors where a large value would assign many neighbors. We set the angle to be half of its maximum within the train set, and hypothesize that most of neighbors within the region is likely of the same class. Particularly, let $H_i = \{e \in Z : \measuredangle(e_i,e) \leq m_i/2\}$ denote the half-angle region where the maximum angle is $m_i := \max_{e \in Z} \measuredangle(e_i,e)$ and the set Z contains all train embeddings. Then, the adaptive angle α_i is defined to contain p-th percentile of the half-angle region:

$$\alpha_i = \measuredangle(e_i, e_{(p \cdot |H_i|)}), \tag{6}$$

where the index (j) of $e_{(j)}$ is sorted in the increasing order of the angle $\measuredangle(e_i,e_{(j)})$. We set $p = 0.85$ in all experiments unless specified otherwise.

After setting the adaptive angle α_i for each train embedding e_i, we estimate its class size based on neighborhoods of neighborhoods. Let $N_\alpha(e_i) = \{e \in Z : \measuredangle(e_i,e) \leq \alpha\}$ denote the neighborhood of e_i, which is the set of all train embedding e within the angle α of e_i. Then, the class size is estimated by the mode of the sizes of neighborhoods of the neighborhood:

$$\kappa_i = \underset{e \in N_{\alpha_i}(e_i)}{\text{mode}} (|N_{\alpha(e)}(e)|), \tag{7}$$

where $\alpha(e)$ is the adaptive angle with respect to the embedding e belonging to the neighborhood $N_{\alpha_i}(e_i)$ of embedding e_i. Using the neighborhoods of neighborhood and mode on it gives more robust estimation of the class size than mere count of the direct neighborhood of e_i.

The proposition in Supp. 9 shows that the angle of neighborhood given by Eq. (6) corresponds to the decision boundary that maximizes the inter-class separation under regularities.

5.1.2. Estimation of maximum size of tail class

After estimating the class size of each training sample by κ_i, we determine the maximum size for the tail classes based on the elbow technique [36]. In particular, by utilizing the sample-wise class sizes κ_i, we first obtain rough estimation of the size of each class set $\eta_y \approx |C_y|$. Then, we estimate the maximum size of tail classes K_{max} by finding a point where the class size most abruptly changes. This optimal shift point is found by the elbow technique. Concretely, we first estimate the size of each class y, $\eta_y \approx |C_y|$, inductively by

$$\begin{aligned} \eta_1 &= round\left(\frac{1}{\kappa_{(1)}} \sum_{i=1}^{\kappa_{(1)}} \kappa_{(i)} \right), \\ \eta_{y+1} &= round\left(\frac{1}{\kappa_{(\eta_y+1)}} \sum_{i=\eta_y+1}^{\min(\kappa_{(\eta_y+1)}, |X|)} \kappa_{(i)} \right), \end{aligned} \tag{8}$$

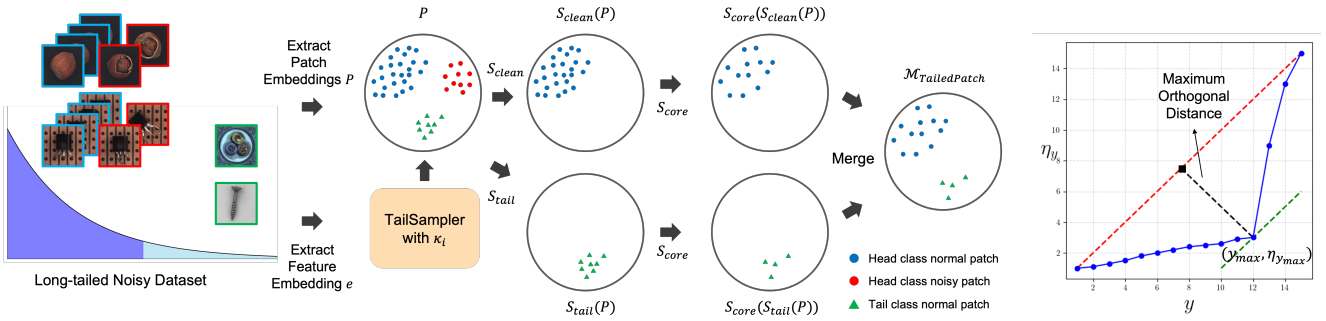
where the index (i) of $\kappa_{(i)}$ is sorted in the increasing order $\kappa_{(1)} \leq \cdots \leq \kappa_{(|X|)}$.

After we acquire the set $\{(y,\eta_y) : y \in (1, \ldots, |\mathcal{Y}|)\}$ through the inductive process above, we define the maximum size of tail classes as the elbow point of the estimated sizes of classes η_y since at this point the class size increases abruptly. The elbow is computed by linearly connecting $(1,\eta_1)$ and $(|\mathcal{Y}|,\eta_{|\mathcal{Y}|})$, and finding the maximum orthogonal distance onto the line from the set of points $\{(y,\eta_y) : y \in (1, \ldots |\mathcal{Y}|)\}$ as shown below:

$$K_{\max} := \eta_{y_{\max}} =: elbow(\eta_1, \ldots, \eta_{|\mathcal{Y}|}) \tag{9}$$

where

$$y_{\max} = \underset{y \in \{1,\ldots,|\mathcal{Y}|\}}{\text{argmax}} \frac{|ay - \eta_y + b|}{\sqrt{a^2+1}} \tag{10}$$



(a) (b)

Figure 3. (a) Sampling process description of TailedCore (**ours**) and (b) the illustration of how we use the elbow method [36].

and where the line $\eta_y = ay + b$ intersects $(1, \eta_1)$ and $(|\mathcal{Y}|, \eta_{|\mathcal{Y}|})$ with $a = (\eta_{|\mathcal{Y}|} - \eta_1)/(|\mathcal{Y}| - 1)$ and $b = \eta_1 - m$. Fig. 3 (b) illustrates Eq. (10). We choose the elbow point $(y_{max}, \eta_{y_{max}})$ as the maximum size of tail classes K_{max}.

The maximum size of tail classes K_{max} allows us to sample the patch features of tail classes exclusively:

$$\mathcal{S}_{tail}(P) = \{\phi_i^{(h,w)} : \kappa_i \leq K_{max}\}. \tag{11}$$

with κ_i given by Eq. (7).

5.2. Proposed method: TailedCore

Our anomaly detection model is built upon the memory bank constructed as below. First, we sample the base memory bank as in [20] with noise-discrimination described in Eq. (3). Then, the base memory is *augmented with TailSampler* $\mathcal{S}_{tail}$ (Eq. (11)). Overall, TailedCore's memory bank is defined by $M_{TailedCore} = \mathcal{S}_{core}(\mathcal{S}_{clean}(P)) \cup \mathcal{S}_{core}(\mathcal{S}_{tail}(P))$, where $\mathcal{S}_{core}$ is the greedy coreset sampling in Eq. (1), $\mathcal{S}_{clean}$ is the noise discrimination process that filters out defect patches based on their outlier scores (Eq. (3)), and the few-shot sampler $\mathcal{S}_{tail}$ augments the tail class patches exclusively on the noise-discriminated base memory bank. The whole process is shown in Fig 3. As TailedCore's patch memory bank is both noise-clean and contains the representative features of few-shot classes, it is robust against both class imbalance and normal sample contamination. In inference, the anomaly score is obtained by Eq. (2) as in the conventional memory-based detector.

6. Experiments

Section 6 consists of dataset description, comparison to the SOTA models, and the detailed ablation study and analysis of TailedCore's components.

6.1. Datasets for unsupervised long-tail noisy anomaly detection

We devise the long-tail noisy anomaly detection datasets by modifying the class-wise sample distributions of the widely used anomaly detection datasets MVTecAD [3] and VisA [44]. We consider three different types of long-tail distributions. Particularly, datasets with step-shaped tail distribution are devised by making the particular classes to have only a small number of samples; namely $K = 1$, and $K = 4$, imposing extreme class imblance on the datasets. We name these two datasets by 'step $(K = 1)$' and 'step $(K = 4)$'. On the other hand, another realistic type of tail-distribution is considered by making the class distribution to follow the Pareto distribution with the shape parameter 0.6. For the step-tailed distribution datasets, we impose 40% of classes to be head classes, containing the original number of training samples. Thus, in MVTecAD, 6 of 15 classes are head classes. For the Pareto tailed distribution, we regard as few-shot (*i.e.*, tail classes) the classes whose number of samples

tail type	Pareto			step ($K=4$)			step ($K=1$)		
class type	C_t	C_h	all	C_t	C_h	all	C_t	C_h	all
PaDiM [9] ICPR'21	82.45	80.95	82.06	77.47	81.28	79.19	71.54	81.75	75.63
HVQ [26] NeurIPS'23	83.46	80.23	82.99	82.01	85.50	83.56	74.15	90.15	80.55
WinCLIP [19] CVPR'23	89.35	90.11	90.37	91.60	88.21	90.37	91.80	88.23	90.37
AnomalyCLIP [43] ICLR'24	90.93	90.98	91.48	91.82	90.83	91.48	91.21	91.90	91.48
PatchCore [34] CVPR'22	93.33	87.59	89.18	92.19	71.18	83.83	86.36	70.48	80.01
SoftPatch [20] NeurIPS'22	84.68	86.95	87.71	67.65	**97.54**	79.64	60.66	**97.49**	75.40
TailedCore (**ours**)	**96.55**	**95.24**	**96.12**	**95.82**	95.34	**95.71**	**93.54**	95.77	**94.43**

Table 1. Anomaly classification on MVTecAD with image-level AUROC (%). We report the mean over 5 random seeds for each measurement. Notations: C_h / C_t: head / tail classes.

tail type	Pareto			step ($K=4$)			step ($K=1$)		
class type	C_t	C_h	all	C_t	C_h	all	C_t	C_h	all
PaDiM [9] ICPR'21	70.70	83.35	78.64	60.65	88.93	72.43	55.98	86.75	68.80
HVQ [26] NeurIPS'23	73.47	84.03	68.25	68.25	89.30	77.02	61.57	80.40	69.42
WinCLIP [19] CVPR'23	73.25	76.92	75.47	75.98	74.76	75.47	78.80	70.80	75.47
AnomalyCLIP [43] ICLR'24	81.96	82.48	82.05	82.28	81.74	82.05	**83.26**	80.34	82.05
PatchCore [34] CVPR'22	86.11	85.73	85.59	83.53	67.51	76.85	79.33	68.56	74.84
SoftPatch [20] NeurIPS'22	78.04	92.16	86.56	59.70	**95.97**	74.81	52.61	**94.17**	69.92
TailedCore (**ours**)	**87.55**	**93.06**	**90.85**	**85.16**	95.91	**89.64**	82.97	94.11	**87.61**

Table 2. Anomaly classification on VisA with image-level AUROC (%). The format and evaluation protocol are the same as Tab. 1.

tail type	Pareto			step ($K=4$)			step ($K=1$)		
class type	C_t	C_h	all	C_t	C_h	all	C_t	C_h	all
PaDiM [9] ICPR'21	90.11	92.66	91.43	82.53	95.29	87.67	78.80	95.54	85.50
HVQ [26] NeurIPS'23	93.63	86.85	90.55	90.73	92.58	91.53	86.36	95.20	89.90
WinCLIP [19] CVPR'23	82.03	84.06	82.29	80.60	84.63	82.29	80.16	85.48	82.29
AnomalyCLIP [43] ICLR'24	91.24	91.69	91.08	89.96	92.66	91.08	89.34	93.68	91.08
PatchCore [34] CVPR'22	93.56	87.98	89.93	93.54	72.09	85.19	92.02	71.35	83.75
SoftPatch [20] NeurIPS'22	92.19	93.83	93.41	80.98	**96.49**	87.24	70.34	**96.89**	80.99
TailedCore (**ours**)	**96.08**	**95.01**	**95.29**	**95.56**	93.20	**94.74**	**94.19**	93.70	**93.99**

Table 3. Anomaly segmentation on MVTecAD with pixel-level AUROC (%). We report the mean over 5 random seeds for each measurement. Notations: C_h / C_t: head / tail classes.

tail type	Pareto			step ($K=4$)			step ($K=1$)		
class type	C_t	C_h	all	C_t	C_h	all	C_t	C_h	all
PaDiM [9] ICPR'21	89.02	95.10	82.81	83.90	97.36	89.51	82.57	96.57	88.40
HVQ [26] NeurIPS'23	95.27	**97.60**	96.71	93.88	**98.34**	95.74	90.58	95.51	92.63
WinCLIP [19] CVPR'23	71.94	73.97	73.19	74.60	71.21	73.19	73.81	72.32	73.19
AnomalyCLIP [43] ICLR'24	95.60	95.46	95.51	95.54	95.48	95.51	**96.16**	94.60	95.51
PatchCore [34] CVPR'22	96.84	87.99	91.13	95.39	62.96	81.88	94.11	65.30	82.10
SoftPatch [20] NeurIPS'22	93.20	96.74	95.27	83.95	97.10	89.43	80.73	96.82	87.43
TailedCore (**ours**)	**97.98**	97.25	**97.48**	**96.80**	97.02	**96.89**	96.12	**97.39**	**96.65**

Table 4. Anomaly segmentation on VisA with pixel-level AUROC (%). The format and evaluation protocol are the same as Tab. 3.

is less than 20 [25]. To impose noisy condition on the datasets, the anomaly samples (*i.e.*, the samples with defective regions) are added to the training samples such that the noisy samples constitute 10% of the training set following the overlap protocol in SoftPatch [20] for fair comparison. In all cases, the testing sets of datasets remain the same as the original MVTecAD and VisA. A more detailed configuration is given in Supp. 11.

6.2. Comparison to the state-of-the-art (SOTA)

We compare TailedCore to the SOTA anomaly detection methods in the unsupervised long-tail noisy settings. We carefully select our baselines. HVQ excels in multi-class learning. PaDiM and SoftPatch are robust against noises as PaDiM is based on Gaussian modeling of the data, while SoftPatch explicitly removes the defect patches by outlier scoring. Using foundation models, WinClip and AnoamlyClip are strong for zero-shot learning. PatchCore are near-perfect in the single-class anomaly detection settings. We show the full experimental results in Supp. 12. All models are trained in the multi-class setting; namely, one model for one dataset with all classes rather than one model per class. The detection performance, however, is measured for each product class under image-level and pixel-level AUROC, and averaged for reporting and comparison. For comparison, we consider models that are specifically robust against noise, strong on multi-class training, and foundation models that can exhibit zero/few-shot learning capability.

Tabs. 1, 2, 3, and 4 show the superiority of TailedCore across all types of long-tail distributions. As anticipated, all baseline models encounter the tail-versus-noise trade-off, resulting in suboptimal performance either on tail classes with limited samples or on head classes with noisy training samples. These trends become more pronounced in highly unbalanced datasets, where the few-shot classes contain only 1 or 4 samples.

6.3. Ablation study and analysis

6.3.1. Tail (few-shot) sampling

We evaluate how accurately our proposed algorithm $\mathcal{S}_{tail}$ can sample tail (few-shot) class samples exclusively without including noise samples. First of all, we measure the ratio of mis-included tail class samples TailSampler which is the ratio of not sampled tail class samples to total tail class samples. A single mis-inclusion can vitally damage anomaly detection performance especially in case of the one-shot setting $K=1$ as the model wouldn't be able to utilize the only sample of the class.

Additionally, the exclusive aspect of few-shot sampling is crucial; the few-shot sampled set shouldn't include anomaly and head class samples as possible. Included anomaly samples would harm the head class performance and head class sampled by few-shot sampler might cause class imbalance issue as well as the possibility of sampling anomaly from head classes. Hence, we measure the ratios of included anomalies and head-class samples along with AUROC which is measured on the outlier scores derived by a given algorithm and ground truth binary labels that indicate whether a given sample is tail class or head class. For TailSampler, the outlier score of each train sample is given as the negative of predicted class size. The included anomaly ratio is defined as the ratio of number of anomaly samples to the number of whole anomaly samples. The ratio of included head-class sample is defined as the number of included head class to the number of included tail class samples.

For extensive comparison, we compare TailSampler with widely used clustering algorithms [2, 13, 16], outlier detectors [1, 5, 24, 35], and density estimators [10, 30]. These methods provide clustering size and/or outlier score/prediction. In cases when only outlier scores are available, we apply the elbow method to induce deterministic prediction of whether a given sample is few-shot or not. We use default settings provided by Sklearn [31] and PyOD [42]. For all methods, we utilize the embedding vectors of WideResNet-50 [40]. Experiments with different architectures and details are provided in Supp. 10.

The combined ratio, which is the sum of missing tail samples and included anomaly samples, indicates that TailSampler is most effective for exclusively sampling few-shot class samples. The most critical factor is to not miss few-shot class samples. On this ratio, TailSampler achieves the lowest rate except for the algorithms that trivially sample all few-shot class samples. We find that most of the classical algorithms for clustering and outlier detection are not usable as they either miss too many few-shot samples or includes a large number of anomaly samples. We note that the class size predictor in our few-shot samples resembles DBSCAN and Affinity Propagation, respectively, in terms of using neighborhoods and voting mechanism. None of these methods however have instance-specific parameters that are adaptively adjusted across different samples.

6.3.2. Class size prediction

We measure how accurately our class size predictor estimates the class size of each sample by measuring the class size prediction error. It is measured for each train sample and weight-averaged through the whole train samples:

$$error = \frac{1}{|\mathcal{X}|} w_i \sum_{i=1}^{|\mathcal{X}|} \left| \kappa_i - |C_{k(i)}| \right| \tag{12}$$

where $k(i)$ indicates the index such that $x_i \in C_{k(i)}$ (namely, $C_{k(i)}$ is the class of x_i). The weight is chosen by $w_i = 1/|C_{k(i)}|$, which is to give more penalty on the few-shot class samples more than the head class samples.

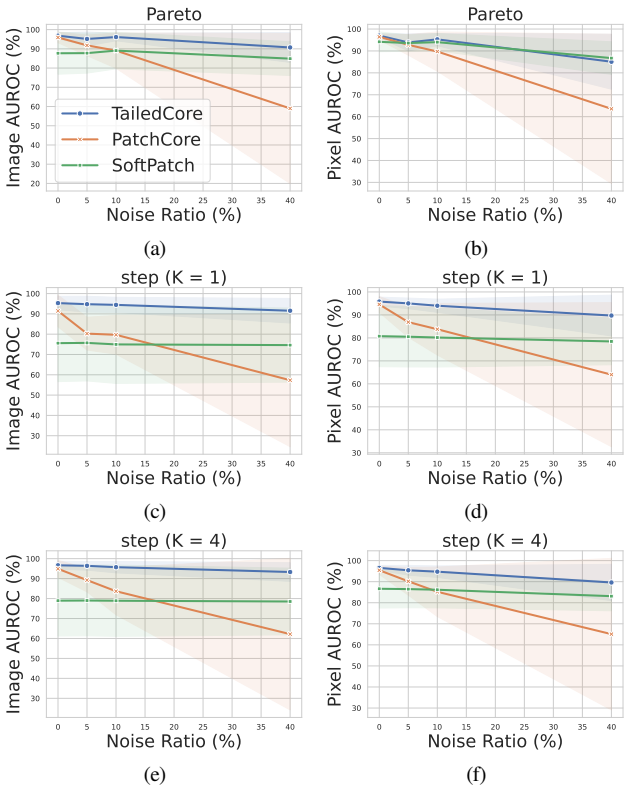


Figure 4. The plot of noise ratio of train data versus anomaly classification (image-level AUROC) and segmentation (pixel-level AUROC) performance. In all cases, TailedCore outperforms both PatchCore and SoftPatch. Dataset is MVTecAD with different tail distributions. The lines show the mean and shades show standard deviation of multiple runs of experiments with different seeds.

method	combined ratio of missing C_t & included AS (↓)	ratio of missing C_t samples (↓)	ratio of included AS (↓)	ratio of included C_h samples (↓)	class size prediction error (↓)	AUROC (%) (↑)
LOF	89.58 ± 44.42	31.22 ± 34.88	58.36 ± 9.54	92.50 ± 7.87	1.00 ± 0.00	77.08 ± 25.86
IF	18.15 ± 15.91	7.25 ± 10.05	10.90 ± 5.86	69.02 ± 25.96	1.00 ± 0.00	98.96 ± 1.27
OCSVM	49.95 ± 9.76	0.78 ± 2.19	49.17 ± 7.57	95.06 ± 6.08	1.00 ± 0.00	96.72 ± 3.59
DBSCAN	35.36 ± 49.55	30.64 ± 44.38	**4.72 ± 5.17**	**25.57 ± 27.24**	0.35 ± 0.44	84.46 ± 22.18
KMeans	94.67 ± 44.95	47.22 ± 35.49	47.45 ± 9.46	97.40 ± 3.99	4.27 ± 1.40	52.58 ± 20.46
GMM	100.00 ± 0.00	**0.00 ± 0.00**	100.00 ± 0.00	97.58 ± 2.86	12.50 ± 1.53	50.00 ± 0.00
KDE	29.81 ± 24.15	2.14 ± 6.26	27.67 ± 17.89	87.47 ± 10.35	1.00 ± 0.00	97.98 ± 2.19
AP	27.81 ± 32.59	12.98 ± 24.01	14.83 ± 8.58	77.45 ± 28.21	0.70 ± 0.22	93.03 ± 12.8
KNN	43.49 ± 31.25	14.46 ± 24.07	29.03 ± 7.18	83.12 ± 18.28	1.00 ± 0.00	94.51 ± 10.34
TailSampler (**ours**)	**5.10 ± 5.07**	0.12 ± 0.65	4.98 ± 4.42	31.96 ± 26.05	**0.19 ± 0.05**	**99.78 ± 0.30**
Ablation on the class size predictor:						
TailSampler (p=100)	8.48 ± 10.88	1.07 ± 5.23	7.41 ± 5.65	42.64 ± 31.86	0.51 ± 0.04	99.76 ± 0.29
TailSampler (p=50)	7.37 ± 10.35	3.46 ± 6.80	3.91 ± 3.55	27.90 ± 27.36	0.12 ± 0.12	97.36 ± 5.65
TailSampler (top-1)	14.47 ± 7.99	0.20 ± 0.78	14.27 ± 7.21	73.97 ± 27.95	0.85 ± 0.04	99.73 ± 0.34
TailSampler (average)	6.48 ± 10.11	3.65 ± 6.99	2.83 ± 3.12	22.16 ± 24.33	0.21 ± 0.07	99.35 ± 2.59

Table 5. The analysis on the proposed TailSampler and its class size predictor. The results are obtained from both MVTecAD and VisA and averaged across all different types of tail distributions and random seeds (overall 60 different seeds) with the standard deviations indicated on the right. The best and second best performances are marked in **bold** and underline, respectively. Acronyms/notations: IF: Isolation Forest; AP: Affinity Propagation; AS: anomaly samples; C_h / C_t: head / tail classes.

task	tail type	Pareto			step ($K=4$)			step ($K=1$)		
	class type	C_t	C_h	all	C_t	C_h	all	C_t	C_h	all
anomaly classification	w/o FSA	82.22	91.22	88.13	60.01	**97.60**	75.05	66.28	**97.85**	78.94
	w/ FSA	**96.55**	**95.24**	**96.13**	**93.54**	95.78	**94.44**	**95.82**	95.34	**95.72**
anomaly segmentation	w/o FSA	91.60	94.94	93.54	68.99	**96.85**	80.14	79.39	**96.35**	86.23
	w/ FSA	**96.09**	**95.01**	**95.30**	**94.19**	93.71	**94.00**	**95.56**	93.20	**94.75**

Table 6. Abaltion on the few-shot augmentation (FSA) of TailedCore. The image-level and pixel-lavel AUROC metrics (%) are reported on the MVTecAD dataset with different types of tail distributions for anomaly classification and segmentation tasks, respectively. Notations: C_h / C_t: head / tail classes.

Tab. 5 shows that most of the clustering based methods vastly fail on this task, and the estimation by our class size predictor outperforms them significantly. We believe this is partly due to suboptimal hyperparameters used for these algorithms. Due to their classical nature, however, it is unclear how to tune them without the availability of the validation set in our scenario.

6.3.3. Abalation study on class size predictor

We conduct detailed ablation study on the proposed class size prediction in Eq. (7). One hyperparameter used in TailSampler S_{tail} of TailedCore is the percentiling proportion p to consider only the majority of nearby samples within the neighborhood. We analyze how this parameter can affect the anomaly detection performance of TailedCore. Tab. 5 indicates that using the nearest 85% embeddings (*i.e.*, $p = 85\%$) is the best choice. Using all of the available neighborhoods (*i.e.*, $p = 100\%$) would include different class samples in the neighborhood, deterioriating class estimation. On the other hand, counting only with the half of samples (*i.e.*, $p = 50\%$) in the neighborhood may lose too much information.

Another component in the class size predictor that requires analysis is its voting mechanism used in Eq. (7). One may simply use the nearest embedding's neighborhood size instead, or the class size of x_i can be estimated by averaging instead of majority vote. As indicated in Tab. 5, both the nearest embedding (*i.e.*, top-1) and averaging are inferior to majority voting. The former is inevitably less robust, while averaging can still be sensitive to the outlier values of neighborhood size.

6.3.4. Ablation on TailedCore

We analyze the impact of augmentation of few-shot class patches by TailSampler in TailedCore. The TailedCore model without these augmented patches has the same memory bank as SoftPatch, but its inference scoring is simpler. Tab. 6 indicates that without the augmented patches from TailSampler, the memory-based model cannot detect anomalies in few-shot class samples. This shows the necessity of tail class augmentation by TailSampler. We note some performance decrease in the head class detection performance since few-shot augmentation may involve few anomaly samples from the head classes that behave like few-shot class instances. This issue intrinsically is due to the weakness of encoder embedding, and the proposed TailSampler based on such embeddings can induce incorrectly sampled few-shot class instances, which we discuss in Sec. 7.

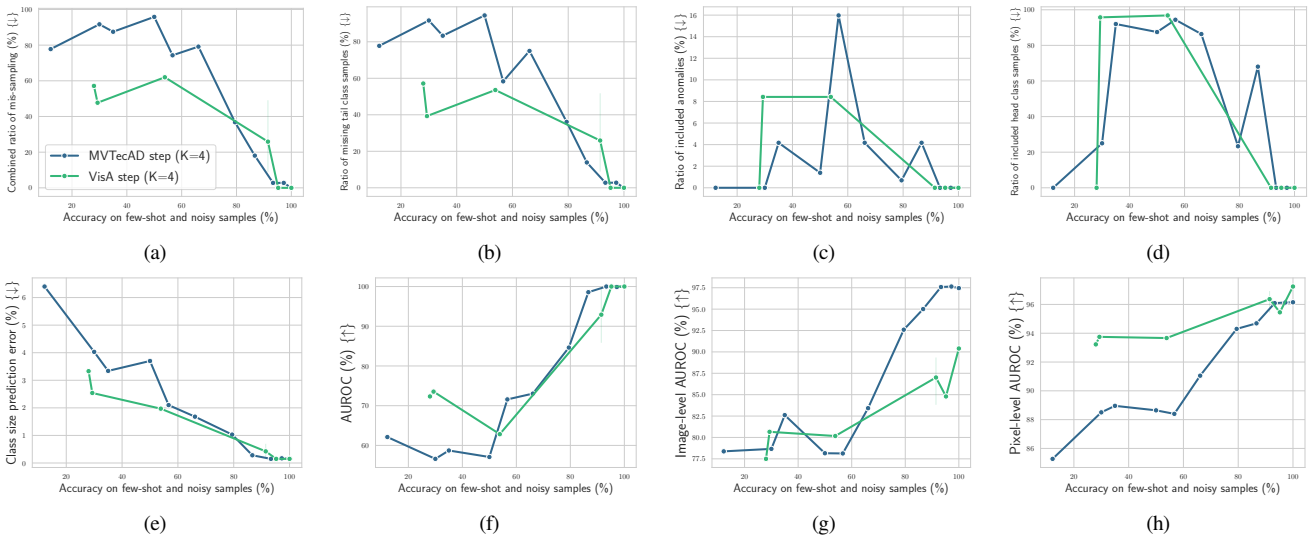


Figure 5. Classification accuracy between tail classes and noisy samples versus metrics relevant to class size prediction and few-shot sampling on MVTecAD and VisA datasets with step-like tailed distribution and $K=4$. The correlation is strong for (a) mis-sampling ratio, (b) ratio of missing few-shot samples, (e) class size prediction error, and (f) AUROC for predicting whether a sample is few-shot or not. We show that improving the discriminative aspect of network's embeddings improves few-shot sampling and prediction error of class size predictor, which in turn improves (g) anomaly classification (image-level AUROC) and (h) anomaly segmentation (pixel-level AUROC) performance of TailedCore.

6.4. Noise ratio

We analyze the impact of noise ratio. Training data with a greater number of defect samples increase the chance of sampling noise instances for the few-shot sampler. We analyze this trend in Fig. 4, but the impact is not severe. In all cases, TailedCore either outperforms or is on par with both PatchCore and SoftPatch.

6.5. Relation between the discriminative aspect of embedding and TailedCore

TailedCore utilizes the embedding features e_i of training samples x_i for few-shot sampling and augmentation thereof. Therefore, TailedCore's detection performance highly relies on the discriminative aspect of the embedding feature. Poorly designed embedding features will not discriminate different class samples in the embedding space. In this case, TailSampler cannot sample tail class samples and it may overly many head class samples. TailedCore's memory bank obtained so will neither be class-balanced nor noise-free, making the inference of TailedCore poor.

In contrast, a highly discriminative embedding features will make TailedCore's class size prediction accurate, thereby enabling it to sample the few-shot class samples exclusively without aggregating any anomaly and head class samples. In which case, TailedCore is both noise-free and class-balanced.

We test this behavior by Oracle labeling and training the embedding features with class labels. Fig. 5 shows that improving discriminative quality of embedding (measured by its classification accuracy on noise and tail class samples) accordingly improves the relevant metrics. Particularly, more discriminative embeddings make more accurate class size prediction, improving TailSampler along with its anomaly classification and segmentation performances.

7. Limitations

The class size prediction in TailSampler can fail if the reflective-symmetric assumption on the inter-class and intra-class similarities breaks down. Particularly, this case can occur if the utilized encoder embedding is either poor or not aligned with the label space of classes. In some cases, the geometric aspect of defect samples highly resembles that of few-shot class instances in the embedding space. We believe that this is due to imperfect nature of encoder embedding, which is often more sensitive to spurious features than the notion of class and object-centric aspect [29]. This issue can be improved by enhancing the encoder network.

8. Conclusion

In this study, we introduced TailedCore, a novel memory-based anomaly detection model designed to effectively address the challenges of unsupervised anomaly detection in noisy and long-tail class distributions. By developing a unique class size predictor and a tailored memory bank, we successfully navigated the tail-versus-noise dilemma, enhancing the model's performance in identifying anomalies within a contaminated and imbalanced dataset. Our comprehensive evaluations demonstrated TailedCore's superior ability to mitigate noise contamination and class imbalance, underscoring its significance in advancing anomaly detection technologies. This work not only showcases a significant stride in handling real-world anomaly detection scenarios but also lays a robust foundation for future research in this critical field.

Acknowledgments Yoon Gyo Jung and Octavia Camps were supported by NSF grant 2038493, ONR grant N00014-21-1-2431, NIH grant R01CA240771 from NCI, and U.S. Department of Homeland Security grant 22STESE00001-03-02. Jaewoo Park and Wonchul Kim were supported by the Technology Innovation Program (20023364) funded by the Ministry of Trade, Industry & Energy (MOTIE, Korea). Kuan-Chuan Peng was exclusively supported by Mitsubishi Electric Research Laboratories. The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the U.S. Department of Homeland Security.

References

[1] Fabrizio Angiulli and Clara Pizzuti. Fast outlier detection in high dimensional spaces. In *European conference on principles of data mining and knowledge discovery*, pages 15–27. Springer, 2002. 3, 6

[2] David Arthur, Sergei Vassilvitskii, et al. k-means++: The advantages of careful seeding. In *Soda*, pages 1027–1035, 2007. 2, 6

[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*, pages 9592–9600, 2019. 5

[4] Paul Bergmann, Kilian Batzner, Michael Fauser, David Sattlegger, and Carsten Steger. Beyond dents and scratches: Logical constraints in unsupervised anomaly detection and localization. *International Journal of Computer Vision*, 130(4):947–969, 2022. 3

[5] 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, 6

[6] Yunkang Cao, Xiaohao Xu, Jiangning Zhang, Yuqi Cheng, Xiaonan Huang, Guansong Pang, and Weiming Shen. A survey on visual anomaly detection: Challenge, approach, and prospect. *arXiv preprint arXiv:2401.16402*, 2024. 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*, pages 383–392, 2022. 2

[8] Zhuohang Chen, Jinglong Chen, Yong Feng, Shen Liu, Tianci Zhang, Kaiyu Zhang, and Wenrong Xiao. Imbalance fault diagnosis under long-tailed distribution: Challenges, solutions and prospects. *Knowledge-Based Systems*, 258:110008, 2022. 1

[9] Thomas Defard, Aleksandr Setkov, Angelique Loesch, and Romaric Audigier. Padim: a patch distribution modeling framework for anomaly detection and localization. In *International Conference on Pattern Recognition*, pages 475–489. Springer, 2021. 2, 5, 6

[10] Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society: series B (methodological)*, 39(1):1–22, 1977. 6

[11] Jan Diers and Christian Pigorsch. A survey of methods for automated quality control based on images. *International Journal of Computer Vision*, pages 1–29, 2023. 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. *arXiv preprint arXiv:2010.11929*, 2020. 2

[13] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *kdd*, pages 226–231, 1996. 2, 6

[14] Chaowei Fang, Lechao Cheng, Huiyan Qi, and Dingwen Zhang. Combating noisy labels in long-tailed image classification. *arXiv preprint arXiv:2209.00273*, 2022. 2

[15] Zheng Fang, Xiaoyang Wang, Haocheng Li, Jiejie Liu, Qiugui Hu, and Jimin Xiao. Fastrecon: Few-shot industrial anomaly detection via fast feature reconstruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 17481–17490, 2023. 2

[16] Brendan J Frey and Delbert Dueck. Clustering by passing messages between data points. *science*, 315(5814):972–976, 2007. 2, 6

[17] Zhaopeng Gu, Bingke Zhu, Guibo Zhu, Yingying Chen, Ming Tang, and Jinqiao Wang. Anomalygpt: Detecting industrial anomalies using large vision-language models. *arXiv preprint arXiv:2308.15366*, 2023. 2

[18] Chaoqin Huang, Haoyan Guan, Aofan Jiang, Ya Zhang, Michael Spratling, and Yan-Feng Wang. Registration based few-shot anomaly detection. In *European Conference on Computer Vision*, pages 303–319. Springer, 2022. 2

[19] Jongheon Jeong, Yang Zou, Taewan Kim, Dongqing Zhang, Avinash Ravichandran, and Onkar Dabeer. Winclip: Zero-/few-shot anomaly classification and segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19606–19616, 2023. 2, 5, 6

[20] Xi Jiang, 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, 5, 6

[21] Justin M Johnson and Taghi M Khoshgoftaar. Survey on deep learning with class imbalance. *Journal of Big Data*, 6(1):1–54, 2019. 1

[22] Yujin Lee, Harin Lim, Seoyoon Jang, and Hyunsoo Yoon. Uniformaly: Towards task-agnostic unified framework for visual anomaly detection. *Available at SSRN 4586132*. 2

[23] Xurui Li, Ziming Huang, Feng Xue, and Yu Zhou. Musc: Zero-shot industrial anomaly classification and segmentation with mutual scoring of the unlabeled images. *arXiv preprint arXiv:2401.16753*, 2024. 2

[24] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. Isolation forest. In *2008 eighth ieee international conference on data mining*, pages 413–422. IEEE, 2008. 3, 6

[25] Ziwei Liu, Zhongqi Miao, Xiaohang Zhan, Jiayun Wang, Boqing Gong, and Stella X Yu. Large-scale long-tailed recognition in an open world. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2537–2546, 2019. 1, 6

[26] Ruiying Lu, YuJie Wu, Long Tian, Dongsheng Wang, Bo Chen, Xiyang Liu, and Ruimin Hu. Hierarchical vector quantized trans-

former for multi-class unsupervised anomaly detection. *Advances in Neural Information Processing Systems*, 36, 2023. 2, 5, 6

[27] Yang Lu, Yiliang Zhang, Bo Han, Yiu-ming Cheung, and Hanzi Wang. Label-noise learning with intrinsically long-tailed data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1369–1378, 2023. 2

[28] Declan McIntosh and Alexandra Branzan Albu. Inter-realization channels: Unsupervised anomaly detection beyond one-class classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6285–6295, 2023. 2

[29] Yifei Ming, Hang Yin, and Yixuan Li. On the impact of spurious correlation for out-of-distribution detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 10051–10059, 2022. 8

[30] Emanuel Parzen. On estimation of a probability density function and mode. *The annals of mathematical statistics*, 33 (3):1065–1076, 1962. 6

[31] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011. 6

[32] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019. 2

[33] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 2

[34] 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*, pages 14318–14328, 2022. 2, 3, 5, 6

[35] Bernhard Schölkopf, Robert C Williamson, Alex Smola, John Shawe-Taylor, and John Platt. Support vector method for novelty detection. *Advances in neural information processing systems*, 12, 1999. 3, 6

[36] Robert L Thorndike. Who belongs in the family? *Psychometrika*, 18(4):267–276, 1953. 4, 5

[37] Tong Wei, Jiang-Xin Shi, Wei-Wei Tu, and Yu-Feng Li. Robust long-tailed learning under label noise. *arXiv preprint arXiv:2108.11569*, 2021. 2

[38] Guoyang Xie, Jinbao Wang, Jiaqi Liu, Feng Zheng, and Yaochu Jin. Pushing the limits of fewshot anomaly detection in industry vision: Graphcore. *arXiv preprint arXiv:2301.12082*, 2023. 2

[39] Zhiyuan You, Lei Cui, Yujun Shen, Kai Yang, Xin Lu, Yu Zheng, and Xinyi Le. A unified model for multi-class anomaly detection. *Advances in Neural Information Processing Systems*, 35:4571–4584, 2022. 1, 2

[40] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016. 6

[41] Manyi Zhang, Xuyang Zhao, Jun Yao, Chun Yuan, and Weiran Huang. When noisy labels meet long tail dilemmas: A representation calibration method. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15890–15900, 2023. 2

[42] Yue Zhao, Zain Nasrullah, and Zheng Li. Pyod: A python toolbox for scalable outlier detection. *Journal of Machine Learning Research*, 20(96):1–7, 2019. 6

[43] Qihang Zhou, Guansong Pang, Yu Tian, Shibo He, and Jiming Chen. AnomalyCLIP: Object-agnostic prompt learning for zero-shot anomaly detection. In *International Conference on Learning Representations*, 2024. 2, 5, 6

[44] Yang Zou, Jongheon Jeong, Latha Pemula, Dongqing Zhang, and Onkar Dabeer. Spot-the-difference self-supervised pre-training for anomaly detection and segmentation. *arXiv preprint arXiv:2207.14315*, 2022. 5

TailedCore: Few-Shot Sampling for Unsupervised Long-Tail Noisy Anomaly Detection

Supplementary Material

9. Supplementary to Method

We state the assumption and proposition from Sec. 5 here in a more rigorous, detailed manner:

Assumption. Let $p_{same}(s|e_i)$ denote the distribution of cosine similarity between e_i and (random vector) embedding e of a same class sample. On the other hand, let $p_{diff}(s|e_i)$ denote the distribution of cosine similarity between e_i and (random vector) embedding e of a different class sample. Then, a reflective symmetry holds between p_{same} and p_{diff} in terms of their angles:

$$p_{same}(\cos(a)|e_i) = p_{diff}(\cos(a_{\max} - a)|e_i). \tag{13}$$

where $a = \arccos(s)$ and $a_{\max} = \arccos(m_i)$. Recall that m_i is the minimum of the support of the inter-class similarity distribution $p_{diff}(\cdot|e_i)$.

Fig. 6 in Supp. 9 shows the feasibility of the above symmetric assumption. We prove Prop. 1 under a reasonable regularity condition, whose trend is evidenced by Fig. 6. Below, we denote by $supp$ the support of a distribution, and for notational brevity, we let $supp(p_{same}) := supp(p(\cdot|e_i))$ and similarly for p_{diff}.

Proposition 1. *For distributions $p_{same}(s|e_i)$ and $p_{diff}(s|e_i)$ that are weakly increasing and decreasing, respectively, let $supp(p_{same})$ be disjoint with $supp(p_{diff})$ and $\max supp(p_{same}) = 1$ Then the threshold given by $\tau_i = \cos(\arccos(m_i)/2)$ separates the supports by*

$$\max supp(p_{diff}) < \tau_i < \min supp(p_{same}), \tag{14}$$

and maximizes the distance between τ_i and $supp(p_{diff}) \cup supp(p_{same})$ under arccosine transformation.

Note that the radius satisfies $r_i = 1 - \tau_i$

Proof. Let $f(a) = p_{same}(\cos a)$ and $g(a) = p_{diff}(\cos a)$ for $0 \le a \le \pi$. Then, by the assumptions from the above, $f(a)$ is decreasing and $g(a)$ is increasing. Since $\max supp(p_{same}) = 1$, we have $\min supp f = 0$. Due to the reflective symmetry, we have

$$f(a) = g(a_{\max} - a), \tag{15}$$

or

$$f(a_{\max}/2 - a) = g(a_{\max}/2 + a). \tag{16}$$

Since $\cos$ is 1-to-1 for the given range of a, disjointness of supports for p_{same} and p_{diff} is preserved for the supports of f and g. Since f is decreasing and g is increasing,

$$a_{\max}/2 \notin supp(f) \cup supp(g), \tag{17}$$

showing that $a_{\max}/2$ separates the supports of f and g. Moreover,

$$a_{\max}/2 = \underset{a}{\operatorname{argmax}} d(a, supp(f) \cup supp(g))) \tag{18}$$

where d is set distance. Observing that

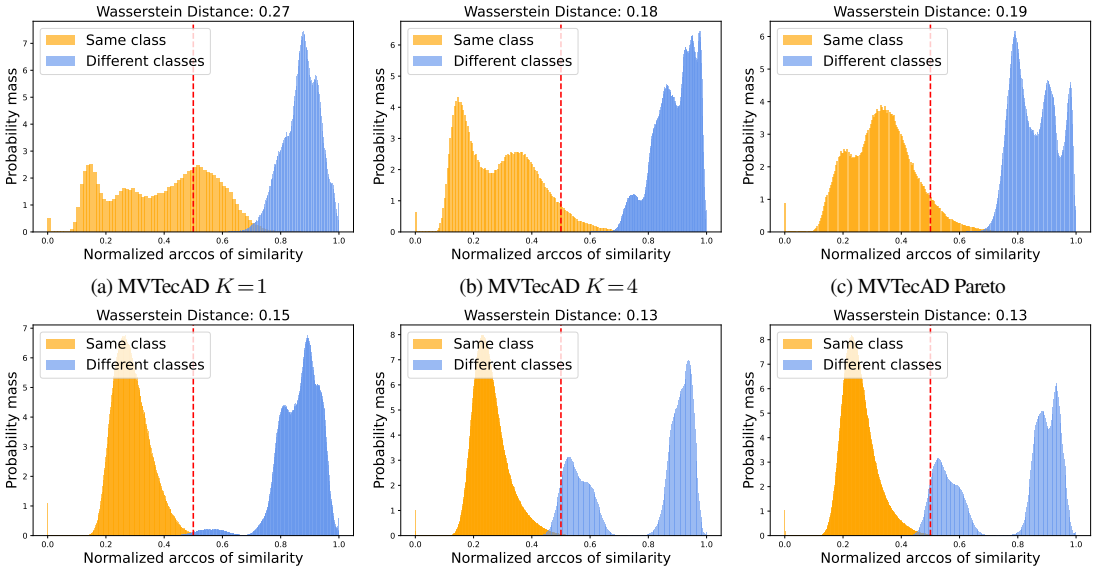
$$a_{\max}/2 = \arccos(m_i)/2 = \arccos(r_i) \tag{19}$$

completes the proof. □

10. Supplementary to Analysis

10.1. Qualitative analysis

Qualitative results of segmentation heatmap are given in Fig. 7, which shores the anomaly scores on the image pixel level. The result shows a trend that TailedCore resolves a performance trade-off between SoftPatch and PatchCore.



(a) MVTecAD $K=1$ (b) MVTecAD $K=4$ (c) MVTecAD Pareto

(d) VisA $K=1$ (e) VisA $K=4$ (f) VisA Pareto

Figure 6. Empirical validation of the assumption in Sec. 5, showing that the inter-class and intra-class similarity distributions approximately exhibit reflective symmetry discussed in the assumption in Sec. 5 and 9

Table 7. Experiments on TailedCore with mix perturbed data in the tail class denoted as TailPert where p is the perturbed ratio, SoftPatch + DBSCAN/K-Means, using transformer's classification token for TailSampler, and noise in tail classes denoted as NoisyTail.

tail type	Pareto			step ($K=4$)			step ($K=1$)		
class type	C_t	C_h	all	C_t	C_h	all	C_t	C_h	all
TailPert $p=0.1$	96.4 / 96.2	95.2 / 94.1	95.9 / 95.2	96.1 / 96.1	95.9 / 93.5	96 / 95.1	93.7 / 93.8	95.4 / 92	94.4 / 93.1
TailPert $p=0.3$	96.2 / 96	95.1 / 94.1	95.8 / 95.1	96.1 / 96.1	95.9 / 93.5	96 / 95.1	93.5 / 93.7	95.7 / 91.9	94.4 / 93
TailPert $p=0.5$	96 / 96	94.5 / 94.9	95.8 / 95.1	95.9 / 96.1	95.6 / 93	95.8 / 94.9	92.5 / 93.3	95.5 / 92	93.7 / 92.8
DBSCAN [13]-SoftPatch [20]	87 / 93.5	93.3 / 95.3	90.6 / 94.4	71.8 / 84.2	97.6 / 96.3	82.2 / 89.1	63.5 / 71.3	97.8 / 96.5	77.2 / 81.4
Kmeans [2]-SoftPatch [20]	86.7 / 92.8	92.5 / 95.1	90.5 / 94.1	71.8 / 83.4	97.6 / 96.4	82.2 / 88.7	63.5 / 71	97.8 / 96.8	77.2 / 81.3
ViT-L [12] ClsToken	96.6 / 96.5	95.3 / 95.3	96.1 / 95.7	96 / 95.5	95.9 / 94.4	96 / 95.2	93.4 / 94.2	96.1 / 95.1	94.5 / 94.6
Noisy Tail	-	-	-	92.3 / 94	95.6 / 92.4	93.7 / 93.5	-	-	-
SoftPatch [20]	84.7 / 92.2	87.0 / 93.8	87.7 / 93.41	67.7 / 81	97.5 / 96.5	79.6 / 87.2	60.7 / 70.3	97.5 / 96.9	75.4 / 81
Ours	96.6 / 96.1	95.2 / 95	96.1 / 95.3	95.8 / 95.6	95.3 / 93.2	95.7 / 94.7	93.5 / 94.2	95.8 / 93.7	94.4 / 94

10.2. Additional experiments and analysis

We conduct additional experiments in order to verify the strengthness of TailedCore with results in Tab. 7. First of all, we perturb the tail class samples with $brightness=0.5$,$constrast=0.5$,$saturation=0.5$,$hue=0.1$ perturbation with the ratio $p\in\{0.1,0.3,0.5\}$. Besides the extreme case where $k=1$, the effect of perturbation is not remarkable showing the robustness of TailedCore.

Secondly, we verify the robustness of TailSampler by substituting it with other unsupervised clustering methods which are DBSCAN [13] and K-Means [2] clustering. For this experiment, we have set the threshold of tail classes to less than 20 samples and normalized with L2 normalization. For hyperparameters, we used $eps=0.5$, $min_samples=2$ for DBSCAN and $k=20$ for KMeans. DBSCAN-SoftPatch sometimes successfully generated 15 clusters+noise class resulting in 16 clusters in total, but often fails and generates more or less clusters. For Kmeans-SoftPatch, we have tried $k=15$, which is the number of classes of MVTec-AD, but this failed detecting any long-tailed clusters. Therefore, we attempted $k=20$ and acquired results as shown in Tab. 7.

Additionally, we substitute CNN with ViT-L [12], specifically the global average pooling feature of CNN with classification token for TailSampler which increased the results slightly as shown in Tab. 7 denoted as "ViT-L ClsToken".

We also test our method when tail classes are contaminated with noise denoted as Noisy Tail. Without the information of classes in an anomalous few-shot scenario, anomalies can't be statistically classified as abnormal from a statistical perspective. Our method tends to specifically address only those instances that are statistically confirmed to be anomalous. Such scenario is tested by including

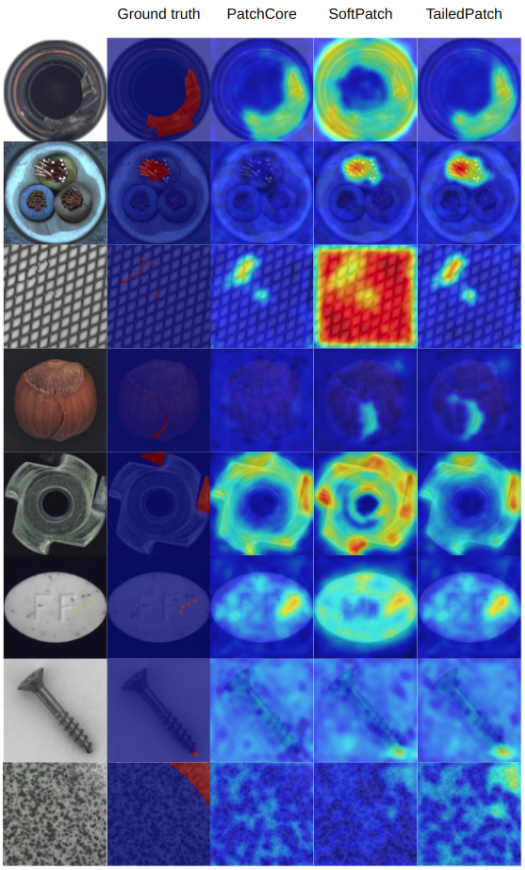


Figure 7. Qualitative analysis of PatchCore, SoftPatch, and TailedCore (**ours**).

Table 8. MVTec-LOCO image AUROC (%) with (all / structural / logical) format.

tail type	Pareto			step ($K=4$)		
class type	C_t	C_h	all	C_t	C_h	all
Patchcore	60.7 / 62.8 / 59.1	59.4 / 76.5 / 45.4	60.2 / 68.3 / 53.6	61.7 / 71.1 / 55.2	63.8 / 70.9 / 59.4	62.5 / 71 / 56.9
SoftPatch	52.3 / 57.8 / 52.9	66.5 / 67.8 / 74.5	58 / 61.8 / 61.6	62.7 / 60.3 / 68	61.7 / 68.4 / 64.3	62.3 / 63.5 / 66.5
Ours	63 / 68 / 59.4	66 / 74.8 / 59.3	64.2 / 70.7 / 59.4	65.9 / 71.7 / 61.7	67.1 / 68.8 / 66.6	66.4 / 70.6 / 63.6

one noisy samples in each of the tail classes with $K=4$ setup (1 noisy sample and 3 normal samples for each tail classes) and the results are shown in Tab. 7. The results show a minor decrease in performance in tail classes, however, the mechanism of our method is fundamentally unable to address such scenarios well.

Finally, we have done experiments on MVTec-Loco [4] dataset to test logical anomaly on PatchCore and our method where results are in Tab. 8. Due to the noisy head class and few-shot tail classes both of the methods do not work well, however, our method still shows better performance comparing with PatchCore.

11. Supplementary to Dataset

Number of total and anomaly samples We provide the detailed number of total and anomaly samples used for our experiments in Tabs. 9, 10, 11, 12, 13, and 14. The names of unspervised long-tail noisy anomaly detection datasets are summarized as follows:

- ***MVTecAD-Pareto***: MVTecAD dataset where its class distribution follows Pareto distribution and 10% of train samples are noises in the head classes
- ***MVTecAD-step-K4***: MVTecAD dataset where its class distribution is imbalanced such that 9 out of 15 classes have only 4 samples for each class and 10% of train samples are noises in the head classes
- ***MVTecAD-step-K1***: MVTecAD dataset where its class distribution is imbalanced such that 9 out of 15 classes have only 1 samples

Table 9. The metadata of MVTecAD-Pareto dataset, indicating the number of total samples in each class and the number of defect samples in each class in the train set.

seed		bottle	cable	capsule	carpet	grid	hazelnut	leather	metal nut	pill	screw	tile	toothbrush	transistor	wood	zipper
101	total	5	5	5	180	297	5	31	22	46	10	69	14	110	5	7
	anomaly	0	0	0	20	33	0	2	2	3	0	4	0	10	0	0
102	total	6	5	5	145	11	5	8	24	59	16	5	40	239	5	86
	anomaly	0	0	0	16	0	0	0	1	7	0	0	5	26	0	5
103	total	231	11	8	5	145	5	24	40	84	6	16	5	5	5	58
	anomaly	22	0	0	0	18	0	1	6	5	0	0	0	0	0	7
104	total	12	38	5	7	5	58	9	5	139	355	86	17	26	5	212
	anomaly	0	3	0	0	0	6	0	0	17	35	8	0	2	0	18
105	total	15	47	5	308	8	22	5	11	5	190	5	6	32	79	122
	anomaly	0	2	0	28	0	1	0	0	0	20	0	0	1	10	16

Table 10. The metadata of MVTecAD-step-K4 dataset, indicating the number of total samples in each class and the number of defect samples in each class in the train set.

seed		bottle	cable	capsule	carpet	grid	hazelnut	leather	metal nut	pill	screw	tile	toothbrush	transistor	wood	zipper
101	total	4	246	241	4	4	430	4	4	4	352	253	66	4	4	4
	anomaly	0	22	22	0	0	39	0	0	0	32	23	6	0	0	0
102	total	4	4	4	4	4	4	4	242	294	352	4	66	4	272	264
	anomaly	0	0	0	0	0	0	0	22	27	32	0	6	0	25	24
103	total	4	4	241	4	290	4	269	4	294	4	4	4	4	272	264
	anomaly	0	0	22	0	26	0	24	0	27	0	0	0	0	25	24
104	total	4	4	4	308	4	430	269	4	4	352	253	4	4	4	264
	anomaly	0	0	0	28	0	39	24	0	0	32	23	0	0	0	24
105	total	230	246	4	4	4	430	269	4	4	352	4	4	234	4	4
	anomaly	21	22	0	0	0	39	24	0	0	32	0	0	21	0	0

for each class and 10% of train samples are noises in the head classes

- ***VisA-Pareto***: VisA dataset where its class distribution follows Pareto distribution and 10% of train samples are noises in the head classes
- ***VisA-step-K4***: VisA dataset where its class distribution is imbalanced such that 7 out of 12 classes have only 4 samples for each class and 10% of train samples are noises in the head classes
- ***VisA-step-K1***: VisA dataset where its class distribution is imbalanced such that 7 out of 12 classes have only 1 samples for each class and 10% of train samples are noises in the head classes

12. Full Experiment Results

The full experiment results for the baseline comparison are given in Tabs. 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, and 26. We note that both WinCLIP and AnomalyCLIP are zero-shot models, hence they do not have performance variance dependant on the train sets (indicated without standard deviation in the tables).

Table 11. The metadata of MVTecAD-step-K1 dataset, indicating the number of total samples in each class and the number of defect samples in each class in the train set.

seed		bottle	cable	capsule	carpet	grid	hazelnut	leather	metal nut	pill	screw	tile	toothbrush	transistor	wood	zipper
101	total	1	1	1	308	1	430	269	1	1	352	253	1	1	1	264
	anomaly	0	0	0	28	0	39	24	0	0	32	23	0	0	0	24
102	total	1	1	241	1	290	1	269	1	294	1	1	1	1	272	264
	anomaly	0	0	22	0	26	0	24	0	27	0	0	0	0	25	24
103	total	1	1	241	1	1	1	269	1	1	352	1	66	234	272	1
	anomaly	0	0	22	0	0	0	24	0	0	32	0	6	21	25	0
104	total	1	1	1	1	290	1	269	242	1	352	1	66	1	1	264
	anomaly	0	0	0	0	26	0	24	22	0	32	0	6	0	0	24
105	total	1	246	241	1	290	430	1	1	294	352	1	1	1	1	1
	anomaly	0	22	22	0	26	39	0	0	27	32	0	0	0	0	0

Table 12. The metadata of VisA-Pareto dataset, indicating the number of total samples in each class and the number of defect samples in each class in the train set.

seed		candle	capsules	cashew	chewinggum	fryum	macaroni1	macaroni2	pcb1	pcb2	pcb3	pcb4	pipe fryum
101	total	290	13	476	9	19	943	29	105	173	68	43	7
	anomaly	14	0	26	0	0	43	1	5	10	4	1	0
102	total	92	5	10	475	53	33	7	256	14	146	21	5
	anomaly	10	0	0	22	2	1	0	11	0	7	0	0
103	total	5	9	86	21	479	253	5	145	7	33	14	52
	anomaly	0	0	4	0	29	10	0	7	0	1	0	2
104	total	170	9	108	65	476	19	30	13	43	954	289	7
	anomaly	6	0	7	1	26	0	2	0	1	49	12	0
105	total	29	101	43	19	177	297	942	13	9	7	67	471
	anomaly	1	1	1	0	14	21	42	0	0	0	3	21

Table 13. The metadata of VisA-step-K4 dataset, indicating the number of total samples in each class and the number of defect samples in each class in the train set.

seed		candle	capsules	cashew	chewinggum	fryum	macaroni1	macaroni2	pcb1	pcb2	pcb3	pcb4	pipe fryum
101	total	4	4	4	4	4	945	4	4	946	950	949	472
	anomaly	0	0	0	0	0	45	0	0	45	45	45	22
102	total	4	4	472	4	4	945	945	4	946	4	4	472
	anomaly	0	0	22	0	0	45	45	0	45	0	0	22
103	total	4	4	4	4	4	4	4	949	946	950	949	472
	anomaly	0	0	0	0	0	0	0	45	45	45	45	22
104	total	945	4	472	476	472	4	4	949	4	4	4	4
	anomaly	45	0	22	23	22	0	0	45	0	0	0	0
105	total	945	4	472	476	4	945	4	949	4	4	4	4
	anomaly	45	0	22	23	0	45	0	45	0	0	0	0

Table 14. The metadata of VisA-step-K1 dataset, indicating the number of total samples in each class and the number of defect samples in each class in the train set.

seed		candle	capsules	cashew	chewinggum	fryum	macaroni1	macaroni2	pcb1	pcb2	pcb3	pcb4	pipe fryum
101	total	1	1	1	1	1	945	1	1	946	950	949	472
	anomaly	0	0	0	0	0	45	0	0	45	45	45	22
102	total	945	1	1	1	1	945	1	1	946	1	949	472
	anomaly	45	0	0	0	0	45	0	0	45	0	45	22
103	total	1	569	472	1	472	945	1	1	946	1	1	1
	anomaly	0	27	22	0	22	45	0	0	45	0	0	0
104	total	1	1	1	1	1	1	1	949	946	950	949	472
	anomaly	0	0	0	0	0	0	0	45	45	45	45	22
105	total	1	569	472	1	472	1	1	949	1	950	1	1
	anomaly	0	27	22	0	22	0	0	45	0	45	0	0

Table 15. Anomaly detection on MVTecAD-Pareto with image-level AUROC. We report the mean over 5 random seeds for each measurement. The best performance is indicated with bold and the second-best with underline.

	PaDiM	SimpleNet	EfficientAD	HVQ	UniAD	WinCLIP	AnomalyCLIP	PatchCore	SoftPatch	TailedCore (ours)
bottle	96.95 ± 2.89	79.89 ± 15.76	97.35 ± 3.28	92.65 ± 9.56	99.30 ± 1.04	98.60	89.30	92.19 ± 13.68	97.95 ± 4.14	**99.76** ± 0.53
cable	66.76 ± 4.43	65.19 ± 9.65	70.05 ± 17.25	70.28 ± 10.73	73.73 ± 12.36	85.00	69.80	**94.81** ± 1.14	78.89 ± 19.04	94.54 ± 2.12
capsule	67.33 ± 7.69	70.59 ± 7.45	69.23 ± 6.25	59.89 ± 2.36	64.25 ± 5.15	68.70	89.90	89.84 ± 3.78	58.78 ± 23.22	**92.19** ± 1.67
carpet	97.51 ± 4.93	78.74 ± 12.14	89.24 ± 10.07	98.48 ± 0.80	98.48 ± 2.41	99.30	**100.00**	86.43 ± 12.97	99.11 ± 0.61	99.26 ± 0.29
grid	64.31 ± 11.57	52.45 ± 2.66	77.61 ± 10.34	78.91 ± 7.71	87.54 ± 9.00	**99.20**	97.00	64.04 ± 17.93	87.10 ± 9.44	91.33 ± 4.74
hazelnut	77.87 ± 21.57	72.94 ± 10.23	95.14 ± 2.48	94.83 ± 3.12	93.66 ± 4.30	92.30	97.20	84.54 ± 25.95	94.26 ± 7.47	**99.28** ± 0.83
leather	99.91 ± 0.14	65.63 ± 15.84	90.34 ± 10.90	99.99 ± 0.02	**100.00** ± 0.00	**100.00**	99.80	99.35 ± 0.97	99.93 ± 0.15	**100.00** ± 0.00
metal nut	71.23 ± 13.34	66.30 ± 13.35	84.15 ± 11.45	87.74 ± 7.61	89.77 ± 15.40	96.20	93.60	96.30 ± 2.50	73.00 ± 20.42	**97.90** ± 1.27
pill	68.33 ± 9.03	72.58 ± 8.61	88.88 ± 7.98	85.07 ± 10.02	81.09 ± 7.31	81.50	81.80	90.25 ± 3.89	90.90 ± 6.37	**93.70** ± 0.93
screw	61.20 ± 2.95	54.47 ± 8.87	71.08 ± 10.29	64.68 ± 14.12	71.07 ± 16.01	71.70	81.10	73.33 ± 8.14	72.04 ± 20.13	**87.24** ± 5.34
tile	94.51 ± 1.22	83.64 ± 15.49	88.59 ± 5.11	94.94 ± 2.07	97.78 ± 1.22	99.90	**100.00**	95.69 ± 3.79	99.02 ± 0.79	97.55 ± 3.25
toothbrush	96.22 ± 1.57	74.11 ± 16.33	94.28 ± 5.92	92.89 ± 4.03	92.44 ± 2.13	85.30	84.70	**99.83** ± 0.37	97.39 ± 3.02	99.50 ± 0.69
transistor	83.27 ± 4.02	64.91 ± 9.85	87.78 ± 8.90	72.44 ± 40.83	90.43 ± 6.33	89.10	92.80	91.62 ± 9.84	86.50 ± 19.44	**96.24** ± 2.23
wood	97.75 ± 1.57	82.05 ± 11.65	**98.61** ± 1.57	78.75 ± 43.50	97.25 ± 0.97	97.60	96.80	94.47 ± 7.58	98.60 ± 0.76	96.47 ± 4.51
zipper	87.99 ± 1.03	78.71 ± 8.46	89.71 ± 5.25	73.44 ± 40.58	90.84 ± 0.24	91.20	98.50	94.58 ± 2.50	82.26 ± 33.79	**96.96** ± 0.62
tail class	82.46 ± 16.66	69.25 ± 12.72	84.30 ± 13.61	83.46 ± 16.17	85.74 ± 14.96	89.36 ± 10.99	90.93 ± 8.42	93.34 ± 9.12	84.69 ± 18.52	**96.55** ± 3.94
head class	80.95 ± 16.24	72.11 ± 13.95	88.73 ± 8.71	80.24 ± 16.46	89.33 ± 7.74	90.11 ± 8.65	90.99 ± 8.61	87.59 ± 15.33	86.96 ± 12.48	**95.24** ± 4.17
all	82.08 ± 16.0	70.81 ± 13.1	86.14 ± 12.0	83.00 ± 18.7	88.51 ± 12.8	90.37 ± 10.1	91.49 ± 8.81	89.82 ± 13.0	87.72 ± 18.0	**96.13** ± 4.28

Table 16. Anomaly detection on MVTecAD-step-K4 with image-level AUROC. We report the mean over 5 random seeds for each measurement. The best performance is indicated with bold and the second-best with underline.

	PaDiM	SimpleNet	EfficientAD	HVQ	UniAD	WinCLIP	AnomalyCLIP	PatchCore	SoftPatch	TailedCore (ours)
bottle	94.30 ± 5.03	74.36 ± 18.09	90.45 ± 10.43	76.78 ± 7.92	96.98 ± 3.02	98.60	89.30	95.49 ± 9.50	65.92 ± 27.24	**99.75** ± 0.26
cable	63.05 ± 5.20	57.45 ± 7.96	75.18 ± 12.58	67.10 ± 21.09	71.47 ± 19.40	85.00	69.80	85.35 ± 9.56	72.85 ± 24.09	**96.18** ± 2.87
capsule	67.41 ± 16.32	62.25 ± 14.52	67.40 ± 27.08	82.37 ± 17.80	69.63 ± 16.31	68.70	89.90	89.18 ± 3.35	64.54 ± 28.56	**93.59** ± 2.67
carpet	99.62 ± 0.39	74.38 ± 8.92	89.07 ± 7.38	98.17 ± 0.83	99.63 ± 0.47	99.30	**100.00**	92.32 ± 13.43	96.90 ± 1.84	99.24 ± 0.48
grid	71.70 ± 3.08	53.85 ± 14.11	66.57 ± 20.92	84.85 ± 3.63	89.49 ± 4.82	**99.20**	97.00	68.60 ± 8.93	74.34 ± 14.00	93.38 ± 3.63
hazelnut	78.01 ± 19.31	71.68 ± 5.07	92.24 ± 7.56	84.18 ± 8.84	92.84 ± 7.23	92.30	**97.20**	63.86 ± 27.00	92.86 ± 9.85	96.67 ± 4.03
leather	99.71 ± 0.65	55.22 ± 16.24	93.43 ± 10.23	98.30 ± 3.75	100.00 ± 0.00	100.00	99.80	85.99 ± 13.06	97.00 ± 6.70	**100.00** ± 0.00
metal nut	64.24 ± 16.45	61.80 ± 14.05	66.98 ± 19.20	74.29 ± 22.95	69.74 ± 17.50	96.20	93.60	88.86 ± 7.32	61.57 ± 22.16	**98.68** ± 0.76
pill	62.20 ± 12.87	62.09 ± 4.64	80.20 ± 14.38	75.99 ± 18.01	73.40 ± 14.43	81.50	81.80	84.16 ± 6.27	72.57 ± 21.56	**93.01** ± 3.00
screw	60.02 ± 1.28	56.22 ± 8.90	80.62 ± 9.46	80.51 ± 11.82	77.02 ± 15.31	71.70	81.10	69.62 ± 8.56	85.28 ± 14.02	**90.14** ± 3.81
tile	86.30 ± 8.87	86.88 ± 5.82	88.89 ± 16.63	89.88 ± 4.80	97.16 ± 0.97	99.90	**100.00**	88.65 ± 14.75	98.69 ± 0.47	94.67 ± 6.50
toothbrush	86.00 ± 7.04	65.39 ± 5.43	81.94 ± 15.66	90.41 ± 8.34	**92.94** ± 1.40	85.30	84.70	85.28 ± 4.96	85.50 ± 13.65	92.44 ± 6.95
transistor	71.84 ± 7.14	60.33 ± 4.80	63.79 ± 17.06	76.06 ± 17.36	76.35 ± 16.41	89.10	92.80	87.47 ± 22.09	59.44 ± 23.09	**95.90** ± 1.56
wood	98.33 ± 0.53	87.98 ± 6.30	98.21 ± 1.18	96.81 ± 2.85	97.37 ± 0.42	97.60	96.80	82.23 ± 22.74	**98.61** ± 0.46	94.53 ± 5.48
zipper	85.25 ± 6.65	73.11 ± 22.83	87.27 ± 11.88	77.78 ± 16.38	91.31 ± 3.99	91.20	**98.50**	91.16 ± 5.59	68.62 ± 39.98	97.62 ± 0.75
tail class	77.47 ± 16.37	65.35 ± 16.11	72.70 ± 14.46	82.01 ± 15.62	81.86 ± 14.30	91.61 ± 9.98	91.82 ± 7.55	92.20 ± 9.62	67.65 ± 19.45	**95.82** ± 3.64
head class	81.29 ± 14.08	69.78 ± 15.08	94.46 ± 4.81	85.51 ± 12.24	92.95 ± 6.18	88.22 ± 11.08	90.84 ± 8.88	71.19 ± 13.95	**97.54** ± 2.98	95.34 ± 4.31
all	79.20 ± 16.79	66.87 ± 14.97	81.48 ± 17.24	83.57 ± 14.74	86.36 ± 14.92	90.37 ± 10.19	91.49 ± 8.81	83.88 ± 15.49	79.65 ± 23.19	**95.72** ± 4.30

Table 17. Anomaly detection on MVTecAD-step-K1 with image-level AUROC. We report the mean over 5 random seeds for each measurement. The best performance is indicated with bold and the second-best with underline.

	PaDiM	SimpleNet	EfficientAD	HVQ	UniAD	WinCLIP	AnomalyCLIP	PatchCore	SoftPatch	TailedCore (ours)
bottle	72.68 ± 15.75	60.40 ± 18.58	57.32 ± 17.54	73.71 ± 13.23	90.74 ± 3.63	98.60	89.30	**99.65** ± 0.12	36.13 ± 3.18	99.25 ± 0.52
cable	57.79 ± 9.61	55.09 ± 10.91	55.21 ± 19.54	58.63 ± 12.98	71.25 ± 18.42	85.00	69.80	76.14 ± 7.38	61.97 ± 20.97	**89.89** ± 4.87
capsule	78.92 ± 17.70	66.29 ± 13.88	65.17 ± 24.08	72.48 ± 19.26	76.86 ± 13.02	68.70	89.90	82.24 ± 4.85	72.55 ± 32.06	**90.39** ± 7.63
carpet	98.53 ± 0.78	57.46 ± 16.18	55.02 ± 17.19	97.88 ± 1.02	99.84 ± 0.10	99.30	**100.00**	92.01 ± 13.37	92.58 ± 6.85	99.13 ± 0.29
grid	67.84 ± 5.25	53.55 ± 12.45	73.28 ± 24.70	83.01 ± 5.46	94.02 ± 4.83	**99.20**	97.00	59.62 ± 7.35	88.69 ± 11.25	91.66 ± 4.84
hazelnut	77.88 ± 18.88	63.34 ± 6.59	69.95 ± 20.92	87.35 ± 7.10	90.43 ± 6.63	92.30	**97.20**	61.91 ± 16.15	84.68 ± 15.04	94.81 ± 4.08
leather	99.14 ± 1.78	54.68 ± 9.26	73.16 ± 23.74	99.99 ± 0.03	99.99 ± 0.02	**100.00**	99.80	79.23 ± 11.61	89.29 ± 23.94	**100.00** ± 0.00
metal_nut	56.36 ± 22.39	59.84 ± 14.35	63.00 ± 19.53	62.08 ± 21.15	63.41 ± 19.72	96.20	93.60	69.07 ± 5.61	55.25 ± 25.76	**96.71** ± 1.97
pill	62.14 ± 12.23	63.15 ± 12.93	73.60 ± 10.82	80.43 ± 17.80	67.64 ± 20.93	81.50	81.80	79.84 ± 2.43	71.72 ± 22.03	**90.51** ± 4.51
screw	61.24 ± 4.09	53.67 ± 10.12	76.44 ± 10.90	79.28 ± 12.84	78.53 ± 15.21	71.70	81.10	67.51 ± 10.18	85.03 ± 15.49	**89.72** ± 7.13
tile	84.31 ± 7.12	76.59 ± 19.53	74.61 ± 21.92	90.08 ± 4.88	97.24 ± 0.77	99.90	**100.00**	93.88 ± 12.51	80.19 ± 11.16	97.66 ± 3.40
toothbrush	84.28 ± 11.86	64.39 ± 8.57	71.78 ± 25.84	87.89 ± 10.45	88.89 ± 4.26	85.30	84.70	83.44 ± 4.81	78.11 ± 20.35	**91.72** ± 7.74
transistor	66.19 ± 9.92	52.65 ± 5.94	57.84 ± 20.07	65.20 ± 15.85	65.44 ± 18.95	89.10	92.80	85.52 ± 22.13	69.04 ± 19.32	**94.08** ± 1.26
wood	98.07 ± 1.04	87.79 ± 3.87	83.37 ± 14.98	95.95 ± 2.94	97.04 ± 1.36	97.60	96.80	81.79 ± 22.33	**98.16** ± 0.73	94.21 ± 6.11
zipper	69.09 ± 20.52	61.23 ± 24.34	60.53 ± 17.06	74.36 ± 18.45	84.42 ± 7.22	91.20	**98.50**	88.32 ± 4.83	67.62 ± 41.47	96.81 ± 1.48
tail class	71.55 ± 19.68	62.35 ± 14.91	59.60 ± 14.40	74.15 ± 18.62	79.06 ± 19.34	91.80 ± 8.38	91.21 ± 8.80	86.36 ± 13.56	60.67 ± 21.77	**93.54** ± 6.19
head class	81.76 ± 14.47	61.49 ± 15.61	78.99 ± 7.19	90.16 ± 8.08	92.37 ± 7.35	88.23 ± 11.75	91.90 ± 7.96	70.48 ± 13.65	**97.50** ± 2.81	95.78 ± 3.57
all	75.63 ± 18.71	62.01 ± 14.95	67.35 ± 16.00	80.55 ± 17.21	84.38 ± 16.80	90.37 ± 10.19	91.49 ± 8.81	80.01 ± 15.56	75.40 ± 24.98	**94.44** ± 5.44

Table 18. Anomaly segmentation on MVTecAD-Pareto with pixel-level AUROC. We report the mean over 5 random seeds for each measurement. The best performance is indicated with bold and the second-best with underline.

	PaDiM	SimpleNet	EfficientAD	HVQ	UniAD	WinCLIP	AnomalyCLIP	PatchCore	SoftPatch	TailedCore (ours)
bottle	95.65 ± 1.24	75.95 ± 11.28	95.66 ± 2.28	96.24 ± 2.71	**97.71** ± 0.40	85.70	90.40	91.32 ± 12.81	97.66 ± 1.52	97.31 ± 2.64
cable	82.43 ± 6.33	66.77 ± 7.52	87.41 ± 11.64	89.50 ± 3.77	**92.45** ± 2.08	61.30	78.90	90.68 ± 2.55	87.09 ± 9.27	91.43 ± 2.33
capsule	97.14 ± 0.58	82.35 ± 7.93	94.96 ± 2.95	97.02 ± 0.22	96.82 ± 0.74	87.00	95.80	**98.51** ± 0.10	94.72 ± 3.52	98.37 ± 0.09
carpet	96.40 ± 3.98	71.85 ± 16.78	88.95 ± 9.43	98.51 ± 0.39	98.43 ± 0.23	90.90	98.80	85.87 ± 13.89	98.89 ± 0.18	**98.48** ± 0.85
grid	67.11 ± 21.48	35.73 ± 17.39	74.45 ± 11.12	92.70 ± 4.10	92.33 ± 3.01	79.40	**97.30**	64.41 ± 15.35	74.97 ± 27.60	91.09 ± 4.28
hazelnut	94.78 ± 0.88	86.32 ± 5.06	**97.15** ± 0.98	96.36 ± 0.98	95.63 ± 1.26	95.70	97.10	83.29 ± 24.37	95.44 ± 1.56	96.37 ± 4.75
leather	99.08 ± 0.15	68.17 ± 34.40	96.07 ± 3.19	98.89 ± 0.13	99.10 ± 0.08	95.50	98.60	99.19 ± 0.15	99.27 ± 0.04	**99.29** ± 00.03
metal_nut	83.10 ± 3.29	76.23 ± 10.81	**96.20** ± 2.87	88.02 ± 2.80	87.81 ± 7.22	49.30	74.40	91.37 ± 7.93	91.34 ± 6.31	91.78 ± 6.62
pill	91.88 ± 4.09	85.69 ± 13.19	95.09 ± 3.37	93.86 ± 2.08	91.79 ± 2.11	72.70	92.00	94.83 ± 4.41	**96.27** ± 2.30	96.25 ± 1.68
screw	94.53 ± 2.41	82.01 ± 9.86	91.90 ± 3.44	94.26 ± 3.35	93.87 ± 4.09	91.10	**97.50**	85.45 ± 10.93	93.27 ± 2.40	96.88 ± 2.42
tile	87.98 ± 3.36	73.42 ± 19.01	83.39 ± 7.08	89.54 ± 1.30	90.52 ± 1.10	79.10	94.60	90.79 ± 5.63	**95.14** ± 0.29	92.31 ± 5.53
toothbrush	97.76 ± 0.94	88.60 ± 8.44	98.35 ± 0.56	98.06 ± 0.60	98.35 ± 0.10	86.20	91.90	98.60 ± 0.06	98.33 ± 0.46	**98.63** ± 0.09
transistor	93.53 ± 3.46	75.27 ± 8.59	87.79 ± 6.41	75.25 ± 41.63	**95.29** ± 3.76	83.70	71.00	93.10 ± 1.02	90.55 ± 7.66	91.45 ± 4.26
wood	93.02 ± 0.93	78.71 ± 6.40	90.31 ± 0.93	73.47 ± 40.56	93.25 ± 0.37	85.10	**96.50**	90.52 ± 8.61	93.62 ± 1.59	91.24 ± 7.74
zipper	97.06 ± 0.74	82.46 ± 9.96	84.90 ± 4.73	76.69 ± 42.35	95.46 ± 0.88	91.70	91.40	91.04 ± 7.48	94.67 ± 8.52	**98.57** ± 0.10
tail class	90.11 ± 10.96	75.12 ± 16.98	90.93 ± 9.36	93.63 ± 4.55	94.16 ± 4.41	82.04 ± 12.69	91.25 ± 8.33	93.57 ± 8.41	92.20 ± 10.33	**96.09** ± 4.12
head class	92.66 ± 5.94	76.25 ± 15.15	91.48 ± 5.75	86.85 ± 12.87	94.86 ± 3.19	84.07 ± 12.24	91.70 ± 8.88	87.99 ± 13.77	93.84 ± 3.83	**95.01** ± 4.55
all	91.43 ± 9.43	75.30 ± 16.4	90.84 ± 7.93	90.56 ± 10.9	94.59 ± 3.8	82.29 ± 12.8	91.08 ± 8.98	89.93 ± 12.0	93.42 ± 8.36	**95.30** ± 4.72

Table 19. Anomaly segmentation on MVTecAD-step-K4 with pixel-level AUROC. We report the mean over 5 random seeds for each measurement. The best performance is indicated with bold and the second-best with underline.

	PaDiM	SimpleNet	EfficientAD	HVQ	UniAD	WinCLIP	AnomalyCLIP	PatchCore	SoftPatch	TailedCore (ours)
bottle	91.44 ± 5.59	70.98 ± 6.76	91.71 ± 4.77	91.42 ± 4.94	96.66 ± 1.00	85.70	90.40	93.79 ± 10.21	80.67 ± 16.93	**98.44** ± 0.06
cable	76.19 ± 11.95	69.18 ± 8.88	90.60 ± 8.38	83.80 ± 9.04	90.16 ± 5.65	61.30	78.90	83.20 ± 10.16	78.37 ± 17.42	**92.70** ± 3.49
capsule	96.29 ± 1.79	78.90 ± 13.70	90.40 ± 9.88	97.65 ± 2.17	96.55 ± 2.04	87.00	95.80	92.31 ± 8.51	93.48 ± 4.96	**98.66** ± 0.15
carpet	97.26 ± 1.57	63.79 ± 34.82	93.59 ± 1.19	96.59 ± 1.85	98.49 ± 0.16	90.90	98.80	93.03 ± 13.35	98.70 ± 0.35	**99.09** ± 0.11
grid	53.43 ± 26.71	24.65 ± 5.41	68.62 ± 18.87	94.05 ± 7.42	91.61 ± 3.53	79.40	**97.30**	68.15 ± 2.89	52.00 ± 28.50	90.66 ± 5.13
hazelnut	95.35 ± 1.51	81.11 ± 10.20	96.28 ± 0.83	93.45 ± 3.30	95.86 ± 2.80	95.70	**97.10**	66.24 ± 29.23	92.85 ± 2.00	85.22 ± 15.52
leather	99.03 ± 0.26	62.90 ± 31.26	95.48 ± 4.09	98.06 ± 2.32	99.04 ± 0.17	95.50	98.60	82.26 ± 17.30	98.41 ± 2.01	**99.30** ± 0.01
metal_nut	78.49 ± 9.04	62.68 ± 17.13	88.03 ± 8.67	82.39 ± 12.96	79.48 ± 7.81	49.30	74.40	90.06 ± 11.31	84.47 ± 8.36	**96.43** ± 0.62
pill	87.44 ± 8.55	73.37 ± 11.91	**96.26** ± 2.17	89.10 ± 6.17	89.02 ± 4.93	72.70	92.00	90.57 ± 7.83	86.67 ± 10.27	95.06 ± 1.74
screw	95.67 ± 2.53	78.80 ± 22.19	94.50 ± 3.76	96.44 ± 3.14	96.23 ± 3.61	91.10	**97.50**	79.08 ± 7.81	93.66 ± 3.04	95.27 ± 2.21
tile	80.17 ± 8.11	76.29 ± 13.48	83.37 ± 9.64	88.75 ± 5.92	90.37 ± 1.03	79.10	94.60	87.22 ± 11.46	**94.98** ± 0.28	92.69 ± 3.70
toothbrush	93.88 ± 2.92	85.92 ± 8.55	94.73 ± 5.03	95.79 ± 3.04	**97.97** ± 0.20	86.20	91.90	93.48 ± 5.36	93.59 ± 4.59	97.87 ± 0.88
transistor	82.59 ± 10.43	61.63 ± 11.46	83.68 ± 4.75	85.40 ± 9.07	86.19 ± 8.95	83.70	71.00	83.49 ± 20.63	73.59 ± 9.03	**88.52** ± 4.14
wood	92.22 ± 3.52	76.12 ± 7.46	90.53 ± 5.62	92.51 ± 3.66	93.35 ± 0.57	85.10	**96.50**	87.54 ± 9.74	94.34 ± 1.09	92.70 ± 2.93
zipper	95.71 ± 4.08	76.50 ± 5.22	92.30 ± 3.88	87.63 ± 13.74	94.74 ± 2.32	91.70	91.40	87.58 ± 9.68	92.86 ± 8.14	**98.65** ± 0.22
tail class	82.53 ± 14.03	66.28 ± 23.04	86.62 ± 9.31	90.73 ± 8.46	90.90 ± 6.21	80.60 ± 11.80	89.96 ± 8.10	93.55 ± 9.15	80.99 ± 15.18	**95.56** ± 3.85
head class	95.29 ± 3.30	74.68 ± 13.83	94.99 ± 3.94	92.58 ± 6.73	96.17 ± 2.72	84.64 ± 11.11	92.66 ± 6.52	72.09 ± 11.76	**96.49** ± 2.45	93.20 ± 7.56
all	87.68 ± 13.7	69.52 ± 19.6	90.01 ± 9.50	91.54 ± 7.91	93.05 ± 6.4	82.29 ± 12.8	91.08 ± 8.98	85.20 ± 14.4	87.24 ± 15.1	**94.75** ± 5.67

Table 20. Anomaly segmentation on MVTecAD-step-K1 with pixel-level AUROC. We report the mean over 5 random seeds for each measurement. The best performance is indicated with bold and the second-best with underline.

	PaDiM	SimpleNet	EfficientAD	HVQ	UniAD	WinCLIP	AnomalyCLIP	PatchCore	SoftPatch	TailedCore (ours)
bottle	76.52 ± 11.38	65.85 ± 16.14	60.97 ± 16.90	85.23 ± 14.39	94.31 ± 0.57	85.70	90.40	98.13 ± 0.08	50.48 ± 7.86	**98.36** ± 0.04
cable	71.28 ± 16.25	65.36 ± 12.90	75.16 ± 13.19	78.51 ± 7.99	**85.37** ± 7.80	61.30	78.90	76.87 ± 9.24	61.93 ± 19.83	83.29 ± 1.47
capsule	97.17 ± 2.50	83.48 ± 10.86	84.41 ± 15.09	97.14 ± 2.20	97.10 ± 1.86	87.00	95.80	87.59 ± 9.90	94.80 ± 5.59	**98.28** ± 0.77
carpet	95.21 ± 3.11	41.13 ± 37.80	68.31 ± 19.53	97.68 ± 0.59	98.43 ± 0.06	90.90	98.80	93.04 ± 13.33	91.00 ± 11.06	**99.04** ± 0.18
grid	71.65 ± 27.37	31.36 ± 16.26	75.59 ± 18.92	94.20 ± 5.63	92.20 ± 4.56	79.40	**97.30**	62.25 ± 6.55	68.87 ± 37.97	88.13 ± 9.06
hazelnut	94.05 ± 1.74	76.90 ± 4.35	88.40 ± 9.31	94.66 ± 1.97	94.69 ± 2.81	95.70	**97.10**	69.53 ± 32.77	90.52 ± 2.27	86.49 ± 15.74
leather	98.66 ± 1.03	60.45 ± 27.40	80.45 ± 20.05	99.12 ± 0.09	99.06 ± 0.18	95.50	98.60	78.87 ± 13.04	94.94 ± 9.75	**99.31** ± 0.01
metal_nut	73.03 ± 12.51	62.16 ± 6.20	89.41 ± 7.27	76.98 ± 10.03	70.76 ± 13.14	49.30	74.40	87.28 ± 5.31	75.83 ± 12.94	**96.01** ± 1.51
pill	85.75 ± 9.04	77.50 ± 9.98	88.57 ± 8.64	91.63 ± 5.53	86.62 ± 6.40	72.70	92.00	91.95 ± 6.94	86.06 ± 11.24	**95.91** ± 1.57
screw	95.53 ± 2.70	73.06 ± 21.56	83.27 ± 22.25	**96.86** ± 3.24	96.75 ± 3.36	91.10	97.50	77.17 ± 7.14	93.88 ± 3.08	95.49 ± 1.58
tile	74.32 ± 7.08	65.06 ± 32.44	67.70 ± 19.74	86.89 ± 2.37	89.83 ± 1.23	79.10	**94.60**	91.10 ± 9.87	69.74 ± 16.33	94.36 ± 1.84
toothbrush	92.99 ± 5.39	81.29 ± 11.74	88.81 ± 10.87	**97.22** ± 2.03	96.77 ± 2.17	86.20	91.90	89.01 ± 9.30	89.17 ± 8.67	96.59 ± 2.00
transistor	73.86 ± 16.73	64.74 ± 9.95	67.85 ± 11.88	79.97 ± 6.07	78.91 ± 7.96	83.70	71.00	83.26 ± 19.73	67.81 ± 15.07	**89.85** ± 0.76
wood	91.40 ± 4.42	73.93 ± 6.83	78.86 ± 12.15	90.74 ± 1.27	92.84 ± 0.64	85.10	**96.50**	82.93 ± 16.39	89.53 ± 4.79	90.38 ± 5.99
zipper	91.08 ± 7.68	56.81 ± 22.34	69.50 ± 13.27	81.71 ± 18.47	93.01 ± 3.31	91.70	91.40	87.41 ± 9.32	90.42 ± 11.43	**98.48** ± 0.42
tail class	78.80 ± 14.63	65.61 ± 16.44	75.37 ± 12.57	86.37 ± 10.24	87.85 ± 10.30	80.17 ± 13.51	89.34 ± 9.75	92.03 ± 8.45	70.34 ± 17.55	**94.19** ± 5.45
head class	95.55 ± 3.46	64.76 ± 19.08	81.48 ± 6.25	95.20 ± 5.33	96.00 ± 3.03	85.48 ± 9.71	93.69 ± 6.52	71.36 ± 13.13	**96.99** ± 2.13	93.71 ± 7.03
all	85.50 ± 14.2(	65.27 ± 19.11	77.82 ± 12.7[illegible]	89.90 ± 9.86	91.11 ± 9.05	82.29 ± 12.8(	91.08 ± 8.98	83.76 ± 14.9[illegible]	81.00 ± 19.0[illegible]	**94.00** ± 6.50

Table 21. Anomaly detection on VisA-Pareto with image-level AUROC. We report the mean over 5 random seeds for each measurement. The best performance is indicated with bold and the second-best with underline.

	PaDiM	SimpleNet	EfficientAD	HVQ	UniAD	WinCLIP	AnomalyCLIP	PatchCore	SoftPatch	TailedCore (ours)
candle	86.88 ± 7.30	80.80 ± 7.81	90.25 ± 6.25	91.63 ± 3.55	80.41 ± 12.16	94.90	79.30	92.12 ± 3.46	84.04 ± 22.34	**95.95** ± 1.18
capsules	57.52 ± 2.04	68.07 ± 5.44	66.71 ± 6.85	63.95 ± 3.76	55.32 ± 1.55	79.40	81.50	66.08 ± 6.72	62.24 ± 5.23	**69.91** ± 5.36
cashew	85.60 ± 3.49	84.66 ± 5.94	91.66 ± 5.64	87.09 ± 7.38	84.99 ± 9.83	91.20	76.30	89.59 ± 9.87	95.95 ± 0.41	**96.18** ± 0.63
chewinggum	88.61 ± 5.90	91.05 ± 8.40	96.94 ± 3.87	95.58 ± 2.28	96.06 ± 1.04	95.50	97.40	94.27 ± 8.69	**98.61** ± 0.52	97.84 ± 1.04
fryum	80.02 ± 6.04	77.24 ± 6.10	89.89 ± 3.25	84.05 ± 2.88	76.49 ± 3.20	73.60	93.00	83.30 ± 8.74	93.31 ± 1.45	**93.96** ± 0.70
macaroni1	78.52 ± 7.18	77.59 ± 4.87	80.72 ± 9.84	65.83 ± 10.11	70.47 ± 7.12	79.00	87.20	81.91 ± 10.80	89.99 ± 7.90	**92.11** ± 4.16
macaroni2	61.87 ± 6.68	52.30 ± 3.42	60.27 ± 6.57	62.94 ± 8.93	58.14 ± 6.88	67.10	**73.40**	64.89 ± 11.32	56.65 ± 10.27	71.48 ± 2.94
pcb1	81.62 ± 6.95	84.91 ± 3.28	93.68 ± 6.21	83.09 ± 2.29	82.91 ± 1.75	72.10	85.40	90.63 ± 1.82	88.40 ± 8.58	**94.35** ± 1.42
pcb2	72.98 ± 9.37	78.32 ± 3.26	87.44 ± 8.82	76.72 ± 8.26	71.95 ± 6.15	47.00	62.20	88.41 ± 4.06	86.56 ± 4.46	**88.87** ± 4.39
pcb3	76.99 ± 11.80	80.35 ± 5.04	91.12 ± 7.12	72.12 ± 8.79	67.34 ± 8.99	63.90	62.70	83.19 ± 16.29	91.76 ± 3.70	**91.95** ± 3.29
pcb4	86.61 ± 7.03	92.90 ± 2.65	97.54 ± 1.85	94.83 ± 2.85	88.61 ± 8.15	74.20	93.90	97.24 ± 3.05	98.69 ± 1.09	**99.39** ± 0.35
pipe fryum	86.52 ± 6.52	79.18 ± 5.31	84.22 ± 13.52	83.08 ± 9.68	81.92 ± 5.73	67.80	92.40	95.50 ± 5.33	92.55 ± 13.18	**98.32** ± 0.72
tail class	70.71 ± 11.15	76.68 ± 11.91	78.50 ± 11.41	73.48 ± 12.57	69.51 ± 13.24	73.26 ± 14.34	81.97 ± 10.27	86.11 ± 13.42	78.05 ± 16.72	**87.55** ± 11.67
head class	83.35 ± 9.34	80.62 ± 10.10	90.60 ± 10.34	84.03 ± 11.84	80.41 ± 11.63	76.92 ± 13.25	82.48 ± 11.38	85.74 ± 11.63	92.17 ± 8.64	**93.06** ± 7.18
all	78.65 ± 11.8[illegible]	78.95 ± 11.7[illegible]	85.87 ± 13.1[illegible]	68.25 ± 12.9[illegible]	76.22 ± 13.3(	75.48 ± 13.9(	82.06 ± 11.7[illegible]	85.59 ± 13.1[illegible]	86.56 ± 15.2[illegible]	**90.86** ± 10.18

Table 22. Anomaly detection on VisA-step-K4 with image-level AUROC. We report the mean over 5 random seeds for each measurement. The best performance is indicated with bold and the second-best with underline.

	PaDiM	SimpleNet	EfficientAD	HVQ	UniAD	WinCLIP	AnomalyCLIP	PatchCore	SoftPatch	TailedCore (ours)
candle	70.21 ± 21.15	72.35 ± 10.28	67.81 ± 26.77	86.47 ± 9.89	66.55 ± 18.77	**94.90**	79.30	86.94 ± 8.04	66.09 ± 30.70	94.30 ± 2.66
capsules	56.49 ± 1.12	68.11 ± 6.70	62.70 ± 6.64	59.55 ± 2.97	56.11 ± 2.86	**79.40**	81.50	59.88 ± 2.99	58.50 ± 4.02	70.06 ± 2.45
cashew	79.82 ± 15.68	77.46 ± 3.32	91.98 ± 6.17	82.54 ± 11.97	77.47 ± 20.14	91.20	76.30	83.38 ± 9.76	86.08 ± 15.53	**95.42** ± 1.66
chewinggum	75.35 ± 21.02	79.44 ± 3.80	85.94 ± 18.73	93.06 ± 4.91	89.45 ± 6.66	95.50	97.40	89.20 ± 11.90	93.92 ± 5.16	**98.62** ± 0.64
fryum	65.18 ± 12.47	75.26 ± 5.35	75.78 ± 12.23	72.47 ± 7.91	64.78 ± 9.52	**73.60**	93.00	86.46 ± 6.80	50.93 ± 24.47	92.35 ± 1.82
macaroni1	76.63 ± 10.17	69.72 ± 10.58	74.02 ± 24.22	70.72 ± 10.86	68.49 ± 10.40	79.00	87.20	70.15 ± 8.37	78.08 ± 24.37	**88.46** ± 9.69
macaroni2	59.73 ± 4.53	50.93 ± 5.62	63.14 ± 8.68	60.06 ± 7.47	55.81 ± 4.23	67.10	**73.40**	60.19 ± 10.80	53.01 ± 10.33	69.59 ± 3.02
pcb1	81.77 ± 14.82	66.43 ± 11.59	84.89 ± 20.74	80.88 ± 12.72	78.39 ± 12.74	72.10	85.40	67.71 ± 13.11	83.60 ± 18.70	**90.46** ± 9.04
pcb2	74.79 ± 16.86	69.33 ± 5.46	89.89 ± 10.52	79.90 ± 16.16	78.16 ± 5.67	47.00	62.20	67.39 ± 18.48	80.39 ± 19.73	**91.29** ± 4.94
pcb3	69.52 ± 16.09	64.65 ± 3.53	73.58 ± 20.21	72.29 ± 13.30	61.79 ± 14.12	63.90	62.70	74.98 ± 17.86	77.18 ± 18.43	**89.82** ± 6.03
pcb4	72.62 ± 21.77	81.42 ± 8.07	88.28 ± 11.79	80.94 ± 15.02	72.40 ± 21.78	74.20	93.90	88.00 ± 13.16	81.46 ± 16.86	**97.01** ± 2.39
pipe fryum	87.17 ± 10.82	74.24 ± 2.02	86.93 ± 14.47	85.41 ± 13.55	83.03 ± 11.05	67.80	92.40	88.01 ± 8.48	88.53 ± 14.75	**98.38** ± 1.05
tail class	60.65 ± 8.82	70.04 ± 11.19	66.91 ± 13.55	68.25 ± 9.89	61.02 ± 10.69	75.99 ± 12.06	82.29 ± 11.69	**83.53** ± 15.54	59.70 ± 15.73	85.17 ± 11.28
head class	88.94 ± 7.80	71.80 ± 8.51	95.31 ± 5.86	89.30 ± 7.59	85.06 ± 9.54	74.76 ± 12.32	81.74 ± 13.29	67.51 ± 13.09	**95.97** ± 6.17	95.92 ± 6.14
all	72.44 ± 16.6	70.78 ± 10.1	78.75 ± 18.2	77.02 ± 14.5	71.04 ± 15.9	75.48 ± 13.9	82.06 ± 11.7	76.86 ± 15.6	74.81 ± 22.2	**89.65** ± 10.76

Table 23. Anomaly detection on VisA-step-K1 with image-level AUROC. We report the mean over 5 random seeds for each measurement. The best performance is indicated with bold and the second-best with underline.

	PaDiM	SimpleNet	EfficientAD	HVQ	UniAD	WinCLIP	AnomalyCLIP	PatchCore	SoftPatch	TailedCore (ours)
candle	51.53 ± 24.34	66.92 ± 19.26	54.68 ± 28.57	71.00 ± 15.55	74.89 ± 6.83	**94.90**	79.30	87.41 ± 7.11	47.01 ± 28.30	92.33 ± 3.58
capsules	55.24 ± 2.23	58.46 ± 4.42	64.51 ± 14.93	60.77 ± 6.42	53.56 ± 0.83	79.40	**81.50**	54.37 ± 4.14	58.59 ± 5.55	64.33 ± 4.21
cashew	71.51 ± 17.61	69.40 ± 12.10	83.28 ± 13.51	71.55 ± 18.13	71.12 ± 18.80	91.20	76.30	88.78 ± 4.47	75.20 ± 19.42	**93.89** ± 3.03
chewinggum	56.00 ± 10.76	65.94 ± 9.60	55.56 ± 3.73	67.93 ± 7.83	62.56 ± 9.82	95.50	97.40	97.30 ± 0.91	75.18 ± 15.66	**97.97** ± 0.34
fryum	67.91 ± 16.31	72.33 ± 8.71	73.71 ± 20.03	71.31 ± 13.56	64.88 ± 12.91	73.60	**93.00**	82.99 ± 1.80	64.21 ± 28.02	91.84 ± 2.50
macaroni1	74.01 ± 12.30	63.53 ± 15.04	78.65 ± 17.86	66.28 ± 15.48	67.13 ± 12.11	79.00	**87.20**	64.53 ± 6.42	76.81 ± 25.38	84.55 ± 14.14
macaroni2	58.82 ± 5.20	57.12 ± 4.10	45.77 ± 10.89	54.65 ± 3.86	59.91 ± 7.73	67.10	**73.40**	58.12 ± 3.22	40.87 ± 5.62	66.36 ± 3.16
pcb1	69.87 ± 20.89	58.32 ± 19.89	75.86 ± 28.20	75.11 ± 14.25	75.65 ± 12.98	72.10	**85.40**	67.99 ± 21.50	71.23 ± 28.26	84.17 ± 12.48
pcb2	79.57 ± 16.79	66.91 ± 7.51	89.59 ± 16.40	76.68 ± 19.53	79.39 ± 5.20	47.00	62.20	60.50 ± 8.81	85.54 ± 21.53	**90.78** ± 9.14
pcb3	72.57 ± 19.16	60.24 ± 8.39	77.67 ± 25.26	66.13 ± 18.37	66.71 ± 12.91	63.90	62.70	68.29 ± 13.90	81.03 ± 21.23	**90.09** ± 8.63
pcb4	80.92 ± 21.97	78.70 ± 7.67	85.96 ± 21.77	77.61 ± 20.38	80.13 ± 22.19	74.20	93.90	81.99 ± 13.97	86.86 ± 17.42	**97.84** ± 2.35
pipe fryum	87.71 ± 9.87	69.71 ± 6.77	79.58 ± 26.83	74.09 ± 17.64	76.19 ± 19.91	67.80	92.40	85.86 ± 7.74	76.60 ± 31.00	**97.22** ± 2.68
tail class	55.98 ± 10.38	62.18 ± 13.25	56.09 ± 14.72	61.58 ± 7.34	60.31 ± 8.56	78.81 ± 12.99	**83.27** ± 10.94	79.33 ± 17.74	52.61 ± 15.29	82.97 ± 13.40
head class	86.76 ± 8.85	70.46 ± 6.12	**94.43** ± 5.11	80.41 ± 6.04	81.99 ± 10.66	70.81 ± 13.42	80.36 ± 13.54	68.56 ± 11.77	94.17 ± 6.81	94.11 ± 6.71
all	68.81 ± 18.8	65.63 ± 12.2	72.07 ± 22.9	69.43 ± 12.9	69.34 ± 14.5	75.48 ± 13.9	82.06 ± 11.7	74.84 ± 16.1	69.93 ± 24.8	**87.61** ± 12.99

Table 24. Anomaly segmentation on VisA-Pareto with pixel-level AUROC. We report the mean over 5 random seeds for each measurement. The best performance is indicated with bold and the second-best with underline.

	PaDiM	SimpleNet	EfficientAD	HVQ	UniAD	WinCLIP	AnomalyCLIP	PatchCore	SoftPatch	TailedCore (ours)
candle	97.49 ± 1.24	92.06 ± 3.38	95.98 ± 0.88	98.63 ± 0.47	97.77 ± 1.38	87.00	**98.80**	92.98 ± 6.06	93.92 ± 10.70	98.75 ± 0.63
capsules	78.32 ± 5.30	91.73 ± 2.28	95.63 ± 2.66	91.77 ± 2.96	83.58 ± 1.76	80.00	95.00	95.92 ± 1.03	88.76 ± 10.75	**97.68** ± 0.49
cashew	95.78 ± 1.37	95.69 ± 1.05	97.48 ± 1.38	**98.74** ± 0.38	98.50 ± 0.53	84.80	93.80	90.20 ± 11.64	98.02 ± 0.25	98.39 ± 0.22
chewinggum	96.86 ± 1.00	94.28 ± 2.57	98.64 ± 0.50	98.54 ± 0.33	98.75 ± 0.10	95.40	**99.30**	94.27 ± 7.88	98.05 ± 0.21	99.06 ± 0.52
fryum	92.73 ± 1.60	85.64 ± 1.58	92.84 ± 2.57	**97.16** ± 0.30	96.45 ± 1.01	87.70	94.60	82.65 ± 8.10	91.37 ± 0.56	91.61 ± 0.54
macaroni1	97.32 ± 1.49	94.19 ± 3.76	96.89 ± 1.66	97.19 ± 1.25	96.80 ± 0.91	50.30	98.30	83.38 ± 17.64	98.27 ± 0.50	**99.23** ± 0.56
macaroni2	90.94 ± 1.98	80.47 ± 6.45	89.86 ± 1.44	94.19 ± 2.04	90.25 ± 2.21	44.70	**97.60**	86.05 ± 16.68	87.28 ± 5.32	92.95 ± 5.85
pcb1	92.39 ± 7.08	96.22 ± 1.72	99.13 ± 0.97	97.97 ± 0.85	97.75 ± 1.52	38.70	94.10	96.14 ± 4.32	98.08 ± 1.58	**99.75** ± 0.04
pcb2	91.86 ± 3.13	90.36 ± 3.12	**97.96** ± 1.19	95.07 ± 1.97	94.24 ± 2.26	58.70	92.40	95.36 ± 3.61	96.87 ± 0.50	97.24 ± 1.61
pcb3	95.76 ± 3.15	95.15 ± 1.92	98.57 ± 0.99	96.41 ± 1.32	95.60 ± 1.86	76.00	88.40	89.19 ± 16.29	98.11 ± 0.25	**99.00** ± 0.26
pcb4	85.95 ± 3.26	90.16 ± 3.11	**98.27** ± 0.51	95.83 ± 1.10	94.66 ± 1.87	91.40	95.70	92.91 ± 6.69	96.35 ± 0.81	97.44 ± 0.71
pipe fryum	98.36 ± 0.74	97.12 ± 0.60	97.38 ± 1.54	**99.10** ± 0.31	98.92 ± 0.17	83.60	98.20	94.62 ± 9.11	98.19 ± 1.12	98.73 ± 0.08
tail class	89.02 ± 8.14	91.02 ± 5.59	96.42 ± 2.51	95.28 ± 3.24	93.01 ± 5.95	71.95 ± 18.13	95.61 ± 2.79	96.84 ± 1.97	93.20 ± 5.59	**97.98** ± 1.35
head class	95.10 ± 3.88	92.58 ± 5.18	96.74 ± 2.94	**97.60** ± 1.41	96.69 ± 2.67	73.97 ± 19.94	95.46 ± 3.32	88.00 ± 12.00	96.74 ± 3.24	97.26 ± 3.31
all	92.82 ± 6.45	91.92 ± 5.49	96.55 ± 3.02	96.72 ± 2.49	95.27 ± 4.61	73.19 ± 19.68	95.52 ± 3.17	91.14 ± 10.44	95.27 ± 5.20	**97.49** ± 2.92

Table 25. Anomaly segmentation on VisA-step-K4 with pixel-level AUROC. We report the mean over 5 random seeds for each measurement. The best performance is indicated with bold and the second-best with underline.

	PaDiM	SimpleNet	EfficientAD	HVQ	UniAD	WinCLIP	AnomalyCLIP	PatchCore	SoftPatch	TailedCore (ours)
candle	92.17 ± 6.03	85.13 ± 9.19	87.24 ± 7.67	97.78 ± 1.33	95.69 ± 2.65	87.00	**98.80**	82.17 ± 22.34	86.22 ± 15.30	98.56 ± 0.69
capsules	73.37 ± 0.66	83.64 ± 4.54	88.48 ± 2.57	88.94 ± 1.14	82.04 ± 0.41	80.00	95.00	94.47 ± 0.55	67.11 ± 3.69	**96.88** ± 0.39
cashew	93.70 ± 4.97	92.02 ± 2.19	96.45 ± 0.76	**98.07** ± 0.52	96.51 ± 3.14	84.80	93.80	78.26 ± 20.71	94.34 ± 4.59	97.74 ± 0.41
chewinggum	93.51 ± 4.69	81.63 ± 5.60	95.61 ± 3.98	**98.06** ± 0.36	97.88 ± 0.81	95.40	99.30	90.31 ± 10.28	97.50 ± 0.69	99.24 ± 0.20
fryum	85.41 ± 4.52	79.59 ± 5.77	92.88 ± 3.67	**95.65** ± 0.93	92.90 ± 2.43	87.70	94.60	86.15 ± 8.46	82.57 ± 3.28	89.69 ± 1.19
macaroni1	95.43 ± 4.08	88.33 ± 8.63	96.72 ± 2.59	97.34 ± 1.56	95.86 ± 2.37	50.30	98.30	73.56 ± 21.73	96.40 ± 3.07	**99.26** ± 0.53
macaroni2	90.05 ± 2.20	74.04 ± 9.95	91.71 ± 3.82	93.46 ± 2.06	89.73 ± 2.26	44.70	**97.60**	84.67 ± 15.24	85.76 ± 1.78	94.27 ± 4.41
pcb1	90.34 ± 11.11	90.72 ± 1.49	**98.26** ± 2.17	96.59 ± 3.74	94.50 ± 6.37	38.70	94.10	72.82 ± 22.46	90.77 ± 11.79	97.88 ± 1.35
pcb2	91.53 ± 7.91	90.22 ± 2.08	**97.49** ± 2.39	95.36 ± 2.89	94.11 ± 3.61	58.70	92.40	74.16 ± 20.63	93.25 ± 4.02	96.36 ± 1.39
pcb3	91.80 ± 5.56	90.10 ± 2.80	95.73 ± 3.21	95.62 ± 2.34	92.02 ± 4.66	76.00	88.40	78.54 ± 26.58	92.23 ± 6.09	**99.04** ± 0.21
pcb4	78.97 ± 12.27	83.57 ± 1.49	92.44 ± 5.82	93.28 ± 4.15	90.17 ± 6.77	91.40	95.70	78.97 ± 21.25	89.96 ± 5.68	**96.71** ± 0.68
pipe fryum	97.88 ± 1.89	96.29 ± 1.06	97.33 ± 0.84	**98.79** ± 0.61	98.44 ± 0.88	83.60	98.20	88.48 ± 11.11	97.14 ± 1.57	97.15 ± 2.68
tail class	83.90 ± 8.10	82.55 ± 6.66	91.46 ± 4.73	93.89 ± 3.30	90.02 ± 5.05	74.61 ± 19.30	95.54 ± 3.00	95.39 ± 5.00	83.96 ± 9.92	**96.80** ± 2.69
head class	97.36 ± 3.71	91.49 ± 4.60	98.02 ± 1.80	**98.35** ± 1.32	97.94 ± 1.97	71.21 ± 19.32	95.48 ± 3.34	62.96 ± 10.16	97.11 ± 4.18	97.03 ± 3.38
all	89.51 ± 9.18	86.27 ± 7.28	94.20 ± 4.94	95.74 ± 3.39	93.32 ± 5.57	73.19 ± 19.68	95.52 ± 3.17	81.88 ± 18.31	89.44 ± 10.08	**96.90** ± 3.01

Table 26. Anomaly segmentation on VisA-step-K1 with pixel-level AUROC. We report the mean over 5 random seeds for each measurement. The best performance is indicated with bold and the second-best with underline.

	PaDiM	SimpleNet	EfficientAD	HVQ	UniAD	WinCLIP	AnomalyCLIP	PatchCore	SoftPatch	TailedCore (ours)
candle	84.28 ± 8.15	80.29 ± 8.82	76.45 ± 11.25	94.56 ± 3.44	88.46 ± 6.15	87.00	**98.80**	88.20 ± 21.36	76.50 ± 12.90	98.18 ± 0.71
capsules	79.63 ± 10.69	75.87 ± 16.78	87.88 ± 10.46	88.76 ± 9.25	83.30 ± 2.13	80.00	95.00	84.81 ± 10.74	75.43 ± 15.75	**95.28** ± 0.36
cashew	90.16 ± 6.83	85.16 ± 18.54	92.51 ± 5.09	93.86 ± 5.48	93.14 ± 5.38	84.80	93.80	96.75 ± 1.03	91.05 ± 6.64	**97.43** ± 1.13
chewinggum	88.79 ± 2.65	68.19 ± 19.17	83.31 ± 7.37	95.56 ± 1.64	94.26 ± 1.22	95.40	99.30	97.73 ± 0.12	94.14 ± 2.73	**99.31** ± 0.14
fryum	85.49 ± 7.54	75.45 ± 17.82	92.83 ± 0.69	**93.47** ± 3.85	92.28 ± 4.44	87.70	94.60	83.20 ± 6.50	84.26 ± 6.65	89.64 ± 2.28
macaroni1	94.04 ± 5.81	78.33 ± 18.65	94.70 ± 5.55	94.64 ± 3.66	95.82 ± 2.37	50.30	98.30	71.40 ± 21.31	90.90 ± 10.48	**98.64** ± 1.33
macaroni2	88.45 ± 0.73	67.73 ± 5.73	86.12 ± 4.12	90.06 ± 0.96	88.14 ± 2.33	44.70	**97.60**	87.97 ± 0.86	82.66 ± 1.09	93.94 ± 1.64
pcb1	84.27 ± 12.99	89.11 ± 3.28	95.83 ± 3.46	92.32 ± 6.40	90.93 ± 7.47	38.70	94.10	82.00 ± 17.26	84.35 ± 13.83	**99.06** ± 0.60
pcb2	93.85 ± 7.95	88.49 ± 3.79	97.46 ± 3.91	92.26 ± 7.46	95.27 ± 3.13	58.70	92.40	63.48 ± 18.14	92.89 ± 8.14	**96.32** ± 1.81
pcb3	92.12 ± 7.98	85.06 ± 6.99	93.42 ± 7.84	91.28 ± 6.56	92.90 ± 5.57	76.00	88.40	69.94 ± 25.21	92.09 ± 8.55	**98.93** ± 0.48
pcb4	81.85 ± 14.74	78.50 ± 2.33	92.59 ± 8.64	88.04 ± 9.46	91.47 ± 8.21	91.40	**95.70**	70.39 ± 19.28	88.89 ± 9.01	95.24 ± 2.23
pipe fryum	97.96 ± 1.82	93.84 ± 2.79	96.21 ± 2.95	96.87 ± 3.28	97.37 ± 2.42	83.60	**98.20**	89.44 ± 11.46	96.06 ± 3.28	97.86 ± 1.82
tail class	82.58 ± 7.84	74.21 ± 11.63	85.44 ± 7.41	90.59 ± 4.34	88.47 ± 4.44	73.81 ± 21.67	**96.17** ± 2.84	94.11 ± 4.19	80.73 ± 9.48	96.13 ± 3.48
head class	96.57 ± 2.80	89.31 ± 5.31	**98.25** ± 1.46	95.51 ± 1.93	96.81 ± 2.83	72.32 ± 18.68	94.61 ± 3.09	65.30 ± 14.32	96.82 ± 2.44	97.39 ± 2.24
all	88.41 ± 9.54	80.50 ± 12.1(	90.78 ± 8.66	92.64 ± 4.8(	91.95 ± 5.8(	73.19 ± 19.6(	95.52 ± 3.17	82.11 ± 17.6(	87.44 ± 10.9(	**96.65** ± 3.10