

Multi-class Image Anomaly Detection for Practical Applications: Requirements and Robust Solutions

Jaehyuk Heo^a, Pilsung Kang^{a,*}

^a*Department of Industrial Engineering, Seoul National University, Republic of Korea*

Abstract

Recent advances in image anomaly detection have extended unsupervised learning-based models from single-class settings to multi-class frameworks, aiming to improve efficiency in training time and model storage. When a single model is trained to handle multiple classes, it often underperforms compared to class-specific models in terms of per-class detection accuracy. Accordingly, previous studies have primarily focused on narrowing this performance gap. However, the way class information is used, or not used, remains a relatively understudied factor that could influence how detection thresholds are defined in multi-class image anomaly detection. These thresholds, whether class-specific or class-agnostic, significantly affect detection outcomes. In this study, we identify and formalize the requirements that a multi-class image anomaly detection model must satisfy under different conditions, depending on whether class labels are available during training and evaluation. We then re-examine existing methods under these criteria. To meet these challenges, we propose Hierarchical Coreset (HierCore), a novel framework designed to satisfy all defined requirements. HierCore operates effectively even without class labels, leveraging a hierarchical memory bank to estimate class-wise decision criteria for anomaly detection. We empirically validate the applicability and robustness of existing methods and HierCore under four distinct scenarios, determined by the presence or absence of class labels in the training and evaluation phases. The experimental results demonstrate that HierCore consistently meets all requirements and maintains strong, stable performance across all settings, highlighting its practical potential for real-world multi-class anomaly detection tasks.

Keywords: Image anomaly detection, Multi-class anomaly detection,

*Corresponding author

Email addresses: jaehyuk.heo@snu.ac.kr (Jaehyuk Heo), pilsung_kang@snu.ac.kr (Pilsung Kang)

1. Introduction

Image anomaly detection is a computer vision task that involves not only determining whether an image contains an abnormal region but also localizing such regions within the image. Unlike conventional classification tasks, it requires both detection and localisation capabilities. However, the development of effective anomaly detection models is hindered by the limited availability and high variability of abnormal data, which makes supervised learning approaches impractical in many cases. To address this, recent studies have primarily adopted an unsupervised learning paradigm, where models are trained solely on normal data and detect anomalies based on deviations from the learned patterns of normality. This unsupervised image anomaly detection (UIAD) approach has demonstrated strong performance and has been widely applied in domains such as visual inspection [1], medical imaging [2, 3], and video surveillance systems [4, 5, 6].

Recently, the UIAD framework has evolved from the conventional one-class setting, where a separate model is trained for each class, to a multi-class setting, in which a single model handles multiple classes simultaneously [7, 8, 9, 10, 11, 12]. One-class UIAD (OC-UIAD) suffers from a major scalability issue, as model storage and computation increase linearly with the number of classes. To improve scalability, multi-class UIAD (MC-UIAD) approaches have been proposed, aiming to detect anomalies across multiple classes using a single model. However, MC-UIAD methods often show inferior performance compared to OC-UIAD, as the representation learned from aggregated data across different classes tends to be less discriminative. Prior studies have primarily focused on minimizing this performance gap [11].

This study goes beyond the performance trade-off between OC-UIAD and MC-UIAD and instead aims to redefine the requirements of MC-UIAD from a practical deployment perspective. Most existing MC-UIAD research either does not use class labels explicitly during training or implicitly leverages them during evaluation. However, the presence or absence of class labels can significantly affect both the training methodology and the performance of multi class anomaly detection in real-world scenarios.

Figure 1 illustrates four possible scenarios for training and evaluation in MC-UIAD, depending on whether class labels are available. To enable applicability across all these scenarios, we define two key requirements for MC-UIAD:

- **Requirement 1:** The model should be trainable regardless of whether class labels are available.

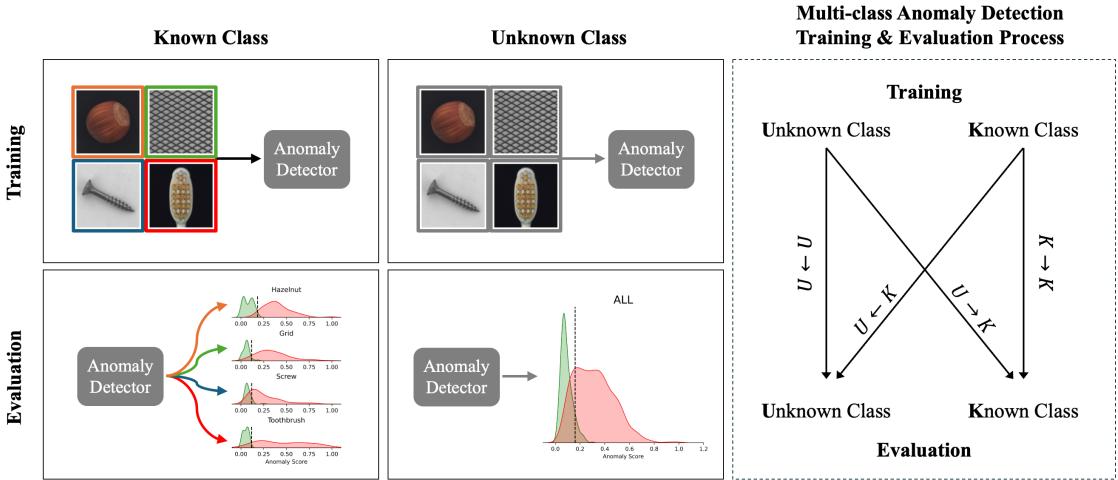


Figure 1: Multi-class image anomaly detection scenarios required for practical applications. The figure presents four combinations of training and evaluation conditions based on the availability of class information: (1) $K \rightarrow K$: training and evaluation are both conducted with known classes, (2) $U \rightarrow K$: the model is trained with unknown classes and evaluated on known classes, (3) $K \rightarrow U$: the model is trained with known classes and evaluated on unknown classes, and (4) $U \rightarrow U$: both training and evaluation are performed without class information. To be practically applicable, an anomaly detection model should be capable of handling all these cases.

- **Requirement 2:** The model should maintain comparable anomaly detection performance during evaluation, irrespective of the availability of class labels.

When class labels are provided, the model can learn separate representations of normal data per class, enabling clearer boundaries between normal and anomalous patterns [11]. In contrast, when class labels are unavailable, the model must learn a shared embedding space for all data. While this may hinder class-wise separation, it has a practical advantage of eliminating annotation costs. Most existing MC-UIAD methods adopt label-agnostic training [7, 8, 9, 10, 12], but as a result, they lack the flexibility to adapt to label-available environments.

Moreover, the presence or absence of class labels during evaluation critically impacts detection performance. Anomaly detection typically requires setting a decision threshold to distinguish between normal and abnormal instances. When class labels are available, optimal class-specific thresholds can be used to maximize detection accuracy. However, in the absence of such labels, a single global threshold must be applied across all classes. Due to inter-class variability, this often results in sub-optimal performance for certain classes, manifesting as increased false positives or false negatives. Figure 2 demonstrates the variation in optimal thresholds at both

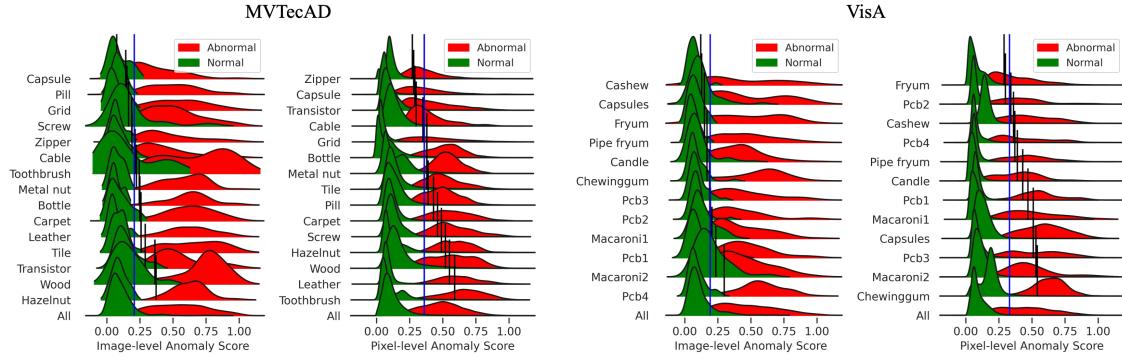


Figure 2: Class-wise and overall anomaly score distributions on MVTecAD and VisA datasets at both image- and pixel-levels. Red and green areas represent distributions of abnormal and normal samples, respectively. Black vertical lines indicate the optimal threshold determined per class, while the blue vertical line indicates the threshold optimized over all classes (i.e., all samples combined).

the image- and pixel-level across classes. Although evaluation performance heavily depends on the presence of class labels, many existing works [12, 13, 14, 15] evaluate models as if class labels were available during inference, even though they were not used during training. This discrepancy highlights the need for a systematic re-evaluation of whether a method can maintain robust performance without class information.

In both training and inference phases, the availability of class labels plays a significant role in MC-UIAD. However, acquiring such labels incurs high annotation costs and may even introduce noise if mislabeling occurs [16]. Therefore, to satisfy the above requirements, it is crucial to design a model capable of distinguishing data distributions and adapting thresholds at a class level without relying on explicit class labels.

To this end, we propose Hierarchical Coreset (HierCore), a novel MC-UIAD framework designed to satisfy both requirements under varying label availability conditions. HierCore operates effectively even without class labels by leveraging semantic information to group data and constructing a hierarchical memory bank using normal samples within each semantic cluster. Specifically, it performs semantic clustering to estimate latent class groupings and assigns cluster-specific keys to build corresponding memory banks. During inference, new inputs are matched to their most relevant semantic cluster, and anomalies are detected based on thresholds defined within each memory bank. We evaluate HierCore across four scenarios defined by the presence or absence of class labels during training and evaluation, and empirically verify its ability to meet all the proposed requirements.

The main contributions of this paper are as follows:

1. We define two requirements that a model must satisfy to enable the practical application of multi-class unsupervised image anomaly detection, depending on whether class information is used during training and evaluation.
2. We systematically re-evaluate existing MC-UIAD methods under four realistic scenarios to assess whether they meet the two defined requirements, using four industrial benchmark datasets.
3. We propose a semantic-aware, hierarchical memory bank framework called HierCore, which remains effective across all scenarios, including those where class labels are unavailable, and achieves robust anomaly detection performance.

The remainder of this paper is structured as follows. Section 2 reviews existing MC-UIAD approaches and highlights limitations in their training and evaluation strategies. Section 3 introduces the proposed HierCore framework and its key components. Section 4 presents the experimental setup and results across the four defined scenarios. Finally, Section 5 concludes the paper and discusses future directions.

2. Related Works

2.1. Multi-class Image Anomaly Detection

MC-UIAD has been developed to enable a single model to detect anomalies across multiple classes, with a focus on improving efficiency in terms of model size and computational cost. Most prior works addressing this problem have employed reconstruction-based approaches that integrate features of normal data from multiple classes into a single model [7, 8, 9, 10, 11, 12]. Reconstruction-based methods learn a model to reconstruct the input image and detect anomalies by calculating the reconstruction error, the difference between the input and its reconstructed version. These approaches assume that a model trained on normal data will fail to accurately reconstruct anomalous inputs. However, they suffer from the *identical shortcut* problem [17], where even anomalous regions can be reconstructed. This problem becomes more severe as the complexity of the normal data increases [12]. To overcome these limitations, several enhancements have been proposed. You et al. [12] introduced UniAD, a Transformer-based model that decomposes input images into patches and applies a neighbor-masked attention module to avoid simply copying neighboring information. Jiang et al. [11] addressed the lack of generalized reconstruction capability across classes by introducing an inter-class inference refinement method that

Table 1: Overview of training and evaluation settings in recent multi-class anomaly detection models. “Unknown” and “Known” indicate whether class labels are assumed to be unavailable or available, respectively. A check mark (✓) denotes that the model is trained or evaluated under the corresponding setting.

Models	Training		Evaluation	
	Unkwown	Known	Unknown	Known
UniAD	✓			✓
ViTAD	✓			✓
InvAD	✓			✓
MambaAD	✓			✓
MINT-AD		✓		✓
HierCore	✓	✓	✓	✓

explicitly leverages class labels. Lu et al. [18] tackled the identical shortcut problem by adopting vector quantization to learn discrete representations of normal data.

An alternative to reconstruction-based methods is the memory bank-based approach [19, 20, 21], which stores features extracted from normal data and compares them with those of new inputs to determine whether they are anomalous. Since these models do not attempt to reconstruct the input, they are not affected by the identical shortcut problem. However, memory usage and computation grow with the number of stored features, posing a scalability challenge.

Our proposed approach inherits the advantages of memory bank-based methods while introducing a hierarchical memory bank structure that improves computational efficiency. This hierarchical design allows us to address both performance and scalability challenges in real-world MC-UIAD scenarios.

2.2. Training and Evaluation of Multi-class Image Anomaly Detection

Table 1 categorizes existing MC-UIAD methods based on whether class information is used during training and evaluation. Most state-of-the-art methods, such as UniAD [12], ViTAD [8], InvAD [9], and MambaAD [7], are designed under the assumption of *unknown-class training*, where class labels are not available during training. As a result, these methods do not leverage class-specific learning strategies. In contrast, MINT-AD [11] assumes *known-class training* and adopts a class-aware architecture that learns representations separately for each class, improving inter-class discrimination. However, such approaches are difficult to apply when class labels are not available. Furthermore, most previous studies assume the availability of class

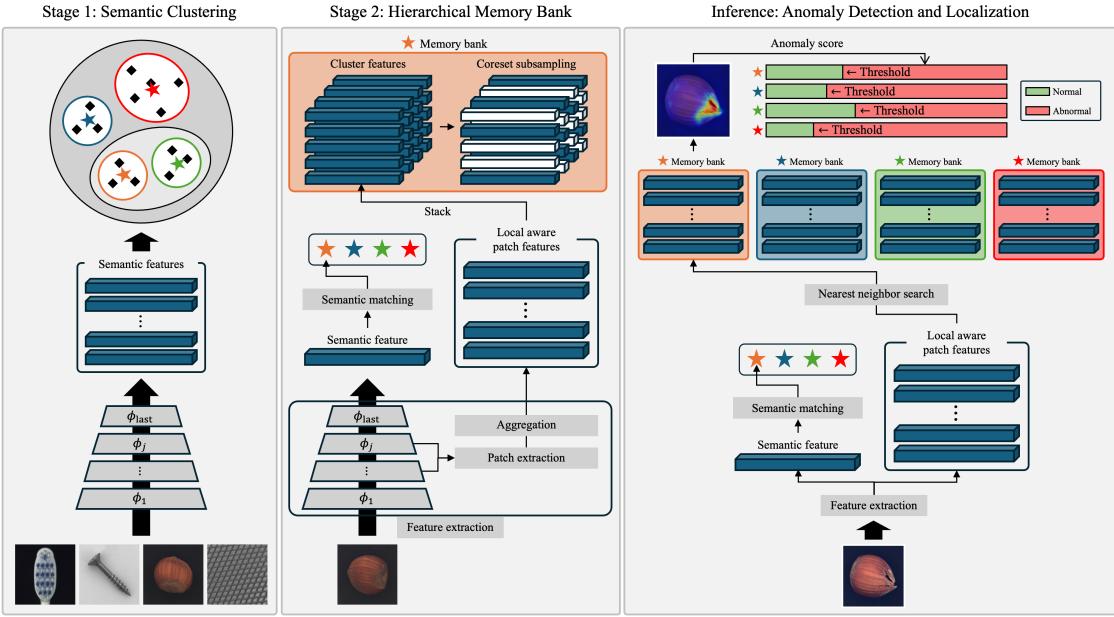


Figure 3: Overview of the HierCore. In Stage 1, semantic features extracted from normal images using a pre-trained encoder are clustered, and the cluster centroids are used as semantic keys for matching. In Stage 2, both semantic features and local patch features are extracted, and patch features are grouped based on the closest semantic key. A coresset-based memory bank is then constructed for each cluster. During inference, features from a test image are matched to the nearest semantic key, and anomaly scores are computed via nearest neighbor search within the corresponding memory bank. Cluster-specific thresholds are used to detect anomalies independent of class labels.

labels during evaluation (*known-class evaluation*), which limits their applicability to real-world scenarios where class labels are often unavailable. This gap highlights a lack of consideration for *unknown-class evaluation* conditions, and the robustness of these models under such settings remains underexplored.

To address this gap, we argue that it is critical to examine whether existing methods can maintain performance under unknown evaluation conditions. We emphasise the importance of developing methods that operate effectively regardless of whether class labels are available during training and evaluation. We also propose HierCore, a novel framework designed to overcome these limitations. HierCore is capable of robust performance even without class information, thanks to its use of a semantic-aware hierarchical memory bank. This structure enables the model to cluster normal data semantically and define a separate anomaly detection threshold for each cluster, even in the absence of explicit class labels.

3. Proposed Method

Figure 3 illustrates the overall architecture and inference procedure of the proposed HierCore framework. HierCore is designed to detect both the presence and location of anomalies without access to class labels. The framework first estimates the latent class of an input image based on its semantic features and then matches the image to the corresponding memory bank. Anomalies are detected by comparing the input features with stored normal patterns in that memory bank.

As shown in Figure 3, HierCore operates in two main stages:

- **Stage 1: Semantic Clustering.** Without access to ground-truth class labels, the semantic embedding of the input image is used to estimate its most likely class, and the image is assigned to the corresponding memory bank (Section 3.1).
- **Stage 2: Hierarchical Memory Bank.** During training, local features of normal images are stored in class-specific memory banks to capture representative normal patterns (Section 3.2).
- **Inference: Anomaly Detection and Localization.** During inference, the estimated class and its corresponding memory bank are used to detect both anomalous regions and the overall anomaly score (Section 3.3).

By constructing a hierarchical memory bank structure based on pseudo-classes, HierCore significantly reduces computational cost compared to single-bank methods such as PatchCore. These efficiency gains are analysed in Section 3.4.

3.1. Semantic Clustering

To enable class-agnostic operation, HierCore estimates pseudo-classes from semantic features rather than relying on explicit labels. This is achieved using a pre-trained image encoder ϕ (e.g., trained on ImageNet [22]), which hierarchically extracts both low-level (edges, colors) and high-level (semantic) features from input images [23]. The final-layer output ϕ_{last} is especially effective at distinguishing images with different semantics within the normal set $\mathcal{X}_N = \{x \mid y = 0\}$. We define the full set of semantic feature vectors \mathbf{e} as:

$$\mathbf{e} = \{\phi_{\text{last}}(x_i) \mid x_i \in \mathcal{X}_N\}.$$

To estimate pseudo-classes, we adopt the FINCH algorithm [24], which clusters semantic embeddings without requiring a predefined number of clusters. FINCH is

computationally efficient, insensitive to hyperparameters, and generates hierarchical clusters. Among the resulting cluster hierarchies, we select the level with the highest Silhouette score [25], and define its cluster count as K .

3.2. Hierarchical Memory Bank

The estimated semantic clusters effectively separate images with distinct patterns, allowing each group to reflect a unique normal distribution. To associate an image with its corresponding cluster, we define a representative key for each cluster as the mean vector of the semantic features contained within that cluster, denoted as $c = \{c_1, c_2, \dots, c_K\}$:

$$c_k = \frac{1}{N_k} \sum_{i=1}^{N_k} e_i, \quad (1)$$

where N_k is the number of normal images assigned to cluster k . Each semantic feature vector e_i from a normal image is assigned to the closest cluster based on the L2 distance d to each cluster's key. The corresponding cluster index k_i^* is defined as follows:

$$k_i^* = \arg \min_{k \in \{1, 2, \dots, K\}} d(e_i, c_k). \quad (2)$$

To detect localized anomalies, we extract local features from intermediate layers $\phi_j(x_i)$ of the backbone network, as in PatchCore [21]. These features are divided into patch-wise representations using a patch extraction function \mathcal{P} , which splits an input feature map into local aware patch features. These are then used to construct a memory bank \mathcal{M}_k for each cluster k :

$$\mathcal{M}_k = \bigcup_{x_i \in \mathcal{X}_{N,k}} \mathcal{P}(\phi_j(x_i)), \quad (3)$$

where $\mathcal{X}_{N,k}$ is the set of normal images assigned to cluster k . Since local features from the same cluster may contain redundant information [21, 26], we apply a coresnet selection algorithm based on k -center clustering to reduce memory size. The final compressed memory bank \mathcal{M}_k^* is defined as:

$$\mathcal{M}_{k,c}^* = \arg \min_{\mathcal{M}_{k,c} \subset \mathcal{M}_k} \max_{m \in \mathcal{M}_k} \min_{n \in \mathcal{M}_{k,c}} \|m - n\|_2, \quad (4)$$

ensuring both representativeness and efficiency.

3.3. Anomaly Detection and Localization

To detect anomalies in a new image, HierCore performs the following steps: (1) Estimate the image’s pseudo-class via semantic embedding; (2) Extract local-aware patch features; (3) Compare each patch with entries in the corresponding memory bank using nearest-neighbor search. The resulting patch-level distances form an anomaly score map, which is upsampled via bilinear interpolation to match the original image resolution. The maximum score in the map is used to determine the image-level anomaly score. This process effectively captures local irregularities while enabling global anomaly detection.

3.4. Computation Costs for Memory Bank

While PatchCore offers efficient inference by storing only normal data without explicit training, it suffers from high memory and computation costs for large or high-resolution datasets [27]. HierCore mitigates this by building class-specific memory banks based on semantic clusters. This reduces both the number of patches and the frequency of feature comparisons. The computational complexity is defined as:

$$\Omega(\text{PatchCore}) = P^2(d + 3r) \quad (5)$$

$$\Omega(\text{HierCore}) = \sum_{i=1}^K P_i^2(d + 3r) + \Omega(N^2), \quad (6)$$

where P is the total number of patches, d is the local feature dimension, r is the sampling ratio for coresset selection, N is the total number of images, and $\Omega(N^2)$ reflects the worst-case complexity of the FINCH algorithm.

Given a stride of 1, the total number of patches is determined by:

$$P = N \cdot (W + 2p - w + 1) \cdot (H + 2p - h + 1),$$

where (W, H) is the image size, (w, h) is the patch size, p is padding. Since $N \ll P$ in typical datasets, the impact of FINCH clustering is negligible in practice.

Furthermore, HierCore achieves faster inference by using fewer coresset entries per cluster, which reduces the number of comparisons in nearest-neighbor search. This approach makes HierCore more efficient and scalable than PatchCore, particularly in multi-class or large-scale settings.

Table 2: Description of four industrial datasets for image anomaly detection.

Class	# Normal		# Abnormal	Anomaly Classes	
	Train	Test	Test		
MVTecAD					
Objects	Bottle	209	20	63	3
	Cable	224	58	92	8
	Capsule	219	23	109	5
	Hazelnut	391	40	70	4
	Metal nut	220	22	93	4
	Pill	267	26	141	7
	Screw	320	41	119	5
	Toothbrush	60	12	30	1
	Transistor	213	60	40	4
	Zipper	240	32	119	7
Textures	Carpet	280	28	89	5
	Grid	264	21	57	5
	Leather	245	32	92	5
	Tile	230	33	84	5
	Wood	247	19	60	5
Total		3,629	467	1,258	-
VisA					
Complex structure	PCB1	904	100	100	4
	PCB2	901	100	100	4
	PCB3	905	101	100	4
	PCB4	904	101	100	7
Multiple instances	Macaroni1	900	100	100	5
	Macaroni2	900	100	100	8
	Capsules	542	60	100	7
	Candles	900	100	100	7
Single instance	Cashew	450	50	100	9
	Chewing gum	453	50	100	6
	Fryum	450	50	100	8
	Pipe fryum	450	50	100	6
Total		8,659	962	1,200	-
MPDD					
Objects	Bracket Black	289	32	47	2
	Bracket Brown	185	26	51	2
	Bracket White	110	30	30	2
	Connector	128	30	14	1
	Metal Plate	54	26	71	3
	Tubes	122	32	69	1
Total		888	176	282	-
BTAD					
Objects	01	400	21	49	1
	02	399	30	200	1
	03	1,000	400	41	1
Total		1,799	451	290	-

4. Experiments

4.1. Experimental Settings

Image Anomaly Detection Benchmarks. To evaluate the performance of the proposed HierCore framework under realistic multi-class conditions, we conduct experiments on four widely used industrial benchmark datasets that encompass a diverse range of classes and anomaly types. Specifically, we use MVTec AD [28], VisA [29], MPDD [30], and BTAD [31], as summarized in Table 2. These datasets pose challenges for multi-class learning due to significant class imbalance, which makes it difficult for a single model to learn and distinguish class-specific characteristics. Furthermore, all datasets except BTAD include multiple types of anomalies per class, requiring models to generalize from normal data representations to various unseen defect types.

Baselines. To comprehensively evaluate HierCore, we compare it against both one-class and MC-UIAD baselines. OC-UIAD models typically achieve higher detection performance on a per-class basis, while MC-UIAD models offer efficiency at the expense of performance. This evaluation aims to quantify the extent to which HierCore improves performance within the MC-UIAD setting, narrowing the gap with OC-UIAD approaches.

The OC-UIAD baselines include: DRAEM [32], SimpleNet [33], RealNet [34], RD [35], RD++ [36], DesTSeg [37], CFLOW-AD [38], PyramidFlow [39], CFA [19], and PatchCore [21]. The MC-UIAD baselines used for comparison are: UniAD [12], DiAD [7], ViTAD [8], InvAD [9], InvAD-lite [9], and MambaAD [10].

Implementation Details. All input images were resized to 256×256 before training and evaluation. For fair comparison, we followed the training protocol used by Zhang et al. [27], training all models for 100 epochs. In HierCore, we used a Wide-ResNet-50 backbone [40] pre-trained on ImageNet-1k. Semantic features were extracted from the fourth layer of the network, while local features were extracted from the second and third layers. Local aware patch features were extracted using a 3×3 sliding window with a stride of 1. The coreset for the memory bank was formed by selecting 10% of all local aware patch features. During inference, the anomaly score for each patch was computed based on the L2 distance to its nearest coresset element in the corresponding memory bank. For efficient nearest neighbor search, we used the GPU-accelerated FAISS library [41]. All experiments were conducted on a system equipped with an Intel i7-8700K CPU, 96GB of RAM, and an NVIDIA GeForce RTX 4090 GPU.

4.2. Evaluation Protocol

We conduct a comprehensive set of experiments to evaluate whether the proposed HierCore satisfies the key requirements of MC-UIAD models in practical scenarios. The evaluation is structured in two main directions: (1) performance comparison between MC-UIAD and OC-UIAD models, and (2) validation of whether MC-UIAD models satisfy the proposed requirements under multi-class settings for real-world deployment.

Evaluating Performance of Image Anomaly Detection in Multi-class Settings. The first set of experiments assesses the performance of HierCore and existing MC-UIAD models under a multi-class setting, comparing them against OC-UIAD baselines. Previous works on MC-UIAD have primarily aimed to minimize performance degradation when transitioning from one-class to multi-class detection. Following this convention, we evaluate detection performance under the following three conditions:

1. OC-UIAD models in one-class setting,
2. OC-UIAD models in multi-class setting,
3. MC-UIAD models in multi-class setting.

Anomaly detection performance is measured at both the image- and pixel-levels, and results are macro-averaged across all classes. For image-level evaluation, we report three metrics: mean Area Under the Receiver Operating Characteristic Curve (mAUROC), mean Average Precision (mAP) [32], and mean F1-score at the optimal threshold (mF1-max) [29]. For pixel-level evaluation, we use five metrics: mAUROC, mAP, mF1-max, mean Area Under the Per-Region-Overlap curve (mAUPRO) [42], and mean Intersection over Union at the optimal threshold (mIoU-max) [9]. In addition, we report an overall score, mean Anomaly Detection score (mAD), defined as the average of all the above metrics.

Evaluating Requirements for Multi-class Image Anomaly Detection. The second set of experiments aims to verify whether HierCore satisfies the two requirements for real-world MC-UIAD, as defined in Section 1. Specifically, we examine the model’s robustness across two representative conditions, based on the availability of class labels during training and evaluation:

- **Condition (1): Unknown → Known & Unknown.** Training without class labels, evaluation under both settings.
- **Condition (2): Known → Known & Unknown.** Training with class labels, evaluation under both settings.

Table 3: Image- and pixel-level mAD on four industrial datasets. The table presents a comparison between one-class and multi-class unsupervised image anomaly detection models. One-class models are evaluated under both one-class and multi-class settings. All experiments are conducted assuming unknown classes during training and known classes during evaluation.

Model	Image-level mAD								Pixel-level mAD								
	MVTecAD	VisA	MPDD	BTAD	MVTecAD	VisA	MPDD	BTAD	MVTecAD	VisA	MPDD	BTAD	MVTecAD	VisA	MPDD	BTAD	
One-class	DRAEM	0.816	0.715	0.743	0.635	0.778	0.538	0.754	0.759	0.183	0.151	0.160	0.101	0.154	0.141	0.182	0.158
	SimpleNet	0.983	0.965	0.936	0.861	0.950	0.894	0.952	0.946	0.664	0.642	0.572	0.547	0.524	0.553	0.582	0.561
	RealNet	0.970	0.899	0.917	0.752	0.792	0.879	0.949	0.926	0.692	0.496	0.540	0.300	0.458	0.511	0.643	0.550
	RD	0.985	0.955	0.949	0.903	0.944	0.922	0.941	0.950	0.688	0.652	0.618	0.591	0.630	0.615	0.672	0.679
	RD++	0.987	0.977	0.943	0.929	0.921	0.913	0.951	0.955	0.695	0.690	0.616	0.620	0.614	0.629	0.672	0.678
	DesTSeg	0.970	0.971	0.904	0.893	0.921	0.908	0.915	0.937	0.731	0.735	0.534	0.549	0.529	0.476	0.547	0.541
	CFLOW-AD	0.961	0.939	0.892	0.867	0.865	0.792	0.947	0.914	0.670	0.623	0.567	0.561	0.564	0.521	0.581	0.598
	PyramidFlow	0.909	0.804	0.892	0.663	0.850	0.767	0.797	0.837	0.605	0.369	0.501	0.284	0.535	0.441	0.523	0.460
	CFA	0.946	0.735	0.868	0.716	0.872	0.850	0.955	0.946	0.624	0.231	0.516	0.395	0.441	0.395	0.648	0.594
	PatchCore	0.992	0.992	0.953	0.953	0.949	0.952	0.952	0.956	0.751	0.745	0.653	0.656	0.656	0.659	0.652	0.653
Multi-class	UniAD	-	0.950	-	0.884	-	0.752	-	0.960	-	0.616	-	0.564	-	0.442	-	0.631
	DiAD	-	0.927	-	0.867	-	0.754	-	0.904	-	0.468	-	0.366	-	0.377	-	0.448
	ViTAD	-	0.980	-	0.895	-	0.881	-	0.950	-	0.691	-	0.577	-	0.582	-	0.665
	InvAD	-	0.983	-	0.944	-	0.934	-	0.961	-	0.700	-	0.628	-	0.632	-	0.691
	InvAD-lite	-	0.980	-	0.936	-	0.911	-	0.950	-	0.690	-	0.611	-	0.610	-	0.676
	MambaAD	-	0.978	-	0.927	-	0.903	-	0.942	-	0.687	-	0.605	-	0.585	-	0.643
	HierCore (Ours)	-	0.992	-	0.953	-	0.948	-	0.952	-	0.748	-	0.649	-	0.655	-	0.652

In Condition (2), which requires models trained with class labels, we exclude MINT-AD [11] due to the lack of a publicly available implementation. Other existing MC-UIAD methods are structurally label-agnostic and do not support training with class labels. As a result, only HierCore is evaluated under this condition.

Since the presence of class labels affects how the decision threshold for anomaly detection is defined, performance may vary between conditions. In this study, we define the optimal threshold \hat{T} as:

$$\hat{T} = \arg \max_T \text{F1-score}(T), \quad \text{based on the precision-recall curve.}$$

Using this criterion, we compare detection performance under each condition to validate whether HierCore maintains stable results regardless of label availability.

4.3. Evaluation of Performance Gap between One-class and Multi-class Image Anomaly Detection

Table 3 presents the performance comparison between OC-UIAD and MC-UIAD models on four industrial datasets: MVTec AD, VisA, MPDD, and BTAD. All evaluations were conducted under the known-class setting, where class information is available during testing.

As expected, OC-UIAD models, which are inherently designed for one-class settings, generally experience performance degradation when extended to multi-class

scenarios. However, PatchCore is a notable exception: despite being an OC-UIAD model, it demonstrates stable and robust performance across both one-class and multi-class settings. This result can be attributed to PatchCore’s memory bank-based architecture, which does not rely on end-to-end training and is not affected by the *identical shortcut* problem that commonly affects reconstruction-based models in multi-class contexts.

While MC-UIAD models tend to outperform OC-UIAD models in multi-class settings, this is not universally the case. For instance, reconstruction-based OC-UIAD models such as RD and RD++ achieve competitive performance compared to several MC-UIAD baselines. PatchCore, although not designed for multi-class settings, achieves top-tier performance on MVTec AD, VisA, and MPDD at both image- and pixel-level evaluations. Its only relative weakness is observed on the BTAD dataset.

The proposed HierCore framework shows a similar trend to PatchCore. Across most datasets (MVTec AD, VisA, MPDD), HierCore outperforms existing MC-UIAD models in both image- and pixel-level anomaly detection. Its performance is particularly strong when class labels are not used during training, further confirming its robustness. On BTAD, while the performance is slightly lower than on other datasets, HierCore remains competitive.

Detailed class-wise performance metrics for all models across the four datasets are provided in Appendix A.

4.4. Evaluation of Requirements for Multi-Class Unsupervised Image Anomaly Detection

4.4.1. Unknown to Known and Unknown

Table 4 shows the performance variation of models trained without class labels under two different evaluation conditions: with and without class label availability. This experiment is designed to assess the robustness of anomaly detection performance in realistic scenarios where class labels may not be available at inference time.

Among the MC-UIAD models, UniAD exhibits a relatively small performance drop under the unknown evaluation condition. However, its overall detection accuracy remains consistently lower than other models, even when class labels are provided at evaluation. This suggests that UniAD’s apparent stability may be due to low baseline performance, rather than true robustness to label absence.

PatchCore achieves performance comparable to HierCore when class labels are available during evaluation. However, it shows a significant performance drop under the unknown evaluation condition. This indicates a strong dependency on class-specific thresholding, which limits its applicability in real-world settings where such

Table 4: Image- and pixel-level anomaly detection performance on four industrial datasets based on average F1-score with optimal thresholding. Models are trained under an unknown classes setting. During evaluation, F1-scores for known classes are computed using class-specific optimal thresholds, while a single optimal threshold is used for unknown-class evaluation. Diff. Ratio indicates the ratio of F1-scores under the unknown-class evaluation to those under the known-class evaluation.

Dataset	Evaluation	Image-level						Pixel-level					
		UniAD	ViTAD	InvAD	MambaAD	PatchCore	HierCore	UniAD	ViTAD	InvAD	MambaAD	PatchCore	HierCore
MVTecAD	Known	91.9	96.0	96.9	96.2	<u>97.8</u>	97.9	48.3	58.9	59.3	57.6	65.4	65.8
	Unknown	87.3	89.2	90.0	88.4	<u>91.7</u>	97.9	43.7	49.0	51.8	51.5	<u>58.7</u>	65.9
	Diff. Ratio	95.1%	92.9%	92.9%	91.9%	93.8%	100.0%	90.4%	83.2%	87.2%	89.4%	89.8%	100.1%
VisA	Known	84.6	85.9	91.5	88.7	92.5	<u>92.2</u>	39.3	41.3	47.2	43.8	52.0	50.9
	Unknown	81.5	81.0	86.3	85.5	<u>90.0</u>	92.2	35.3	34.4	42.3	39.4	<u>45.6</u>	50.9
	Diff. Ratio	96.3%	94.2%	94.3%	96.4%	<u>97.2%</u>	100.0%	90.0%	83.3%	89.7%	90.0%	87.8%	100.0%
MPDD	Known	78.8	84.9	84.9	91.2	92.2	<u>91.8</u>	20.5	37.7	37.7	46.2	50.6	50.3
	Unknown	76.9	77.0	77.0	<u>84.0</u>	83.8	91.7	15.4	22.5	22.5	40.5	28.6	50.5
	Diff. Ratio	97.5%	90.8%	90.8%	92.1%	90.9%	99.9%	75.4%	59.6%	59.6%	<u>87.5%</u>	56.5%	100.5%
BTAD	Known	93.1	92.9	93.9	92.7	<u>93.7</u>	92.8	54.1	<u>56.9</u>	61.4	55.7	56.5	56.5
	Unknown	81.0	77.9	80.3	75.7	<u>78.4</u>	92.8	53.0	22.8	53.3	49.3	<u>55.2</u>	56.5
	Diff. Ratio	87.0%	83.9%	85.5%	81.7%	83.7%	100.0%	98.0%	40.0%	86.8%	88.6%	97.7%	100.0%

labels are not accessible.

In contrast, HierCore maintains high and consistent anomaly detection performance regardless of whether class labels are available during evaluation. This result highlights the reliability of HierCore in scenarios where class information is unavailable, an important property for real-world MC-UIAD applications.

Detailed class-wise performance comparisons for all four datasets are provided in Appendix B.

4.4.2. Known to Known and Unknown

When class labels are available during training, HierCore constructs a separate memory bank for each class. During evaluation, if class labels are also available, each input is directly matched to its corresponding class-specific memory bank, and anomaly detection is performed accordingly. This is functionally similar to deploying a separate PatchCore model for each class.

In contrast, under the unknown evaluation setting, where class labels are not accessible, HierCore estimates the most likely class cluster for a given input using its semantic embedding and the key vector computed as described in Eq. (2). Anomaly detection is then carried out using the corresponding estimated cluster’s memory bank.

Table 5 presents the results of this setting, in which HierCore is trained with known class labels and evaluated both with and without class information. While performance robustness slightly decreases compared to the results in Table 4, HierCore

Table 5: Average F1-score on the optimal threshold. Known training setting. Optimal threshold per class is applied in Known evaluation. Optimal threshold for all samples is applied in Unknown evaluation. Diff. Ratio is the ratio of F1-score in Unknown with respect to Known.

Dataset	Evaluation	HierCore	
		Image-level	Pixel-level
MVTecAD	Known	97.9	66.1
	Unknown	97.9	65.9
	Diff. Ratio	100.0%	99.6%
VisA	Known	92.2	51.3
	Unknown	92.2	50.9
	Diff. Ratio	100.0%	99.2%
MPDD	Known	91.8	50.4
	Unknown	91.8	48.0
	Diff. Ratio	100.0%	95.2%
BTAD	Known	92.8	56.5
	Unknown	92.8	56.5
	Diff. Ratio	100.0%	100.0%

still outperforms other models’ class-aware performance even when evaluated without class labels.

These results suggest that having class labels during training can enhance detection performance by enabling better modeling of class-specific characteristics. However, when comparing Table 4 and Table 5, it becomes evident that the semantically guided memory bank construction used in label-agnostic training leads to more stable and reliable performance across both evaluation conditions. This highlights the practical advantage of HierCore’s semantic clustering approach for handling real-world scenarios where class information may be missing or unreliable.

4.5. Visualization of Semantic Centroids and Test Samples

In this section, we verify the effectiveness of HierCore’s approach of separating normal data based on semantic features and organizing memory banks accordingly. Figure 4 provides a 2D visualization of the clustering results obtained from FINCH, using semantic embeddings of normal samples. We apply t-SNE for dimensionality reduction to reveal the structural grouping of the data.

The visualization reveals an interesting phenomenon: in the MPDD dataset, for instance, normal samples that share the same class label are often distributed across

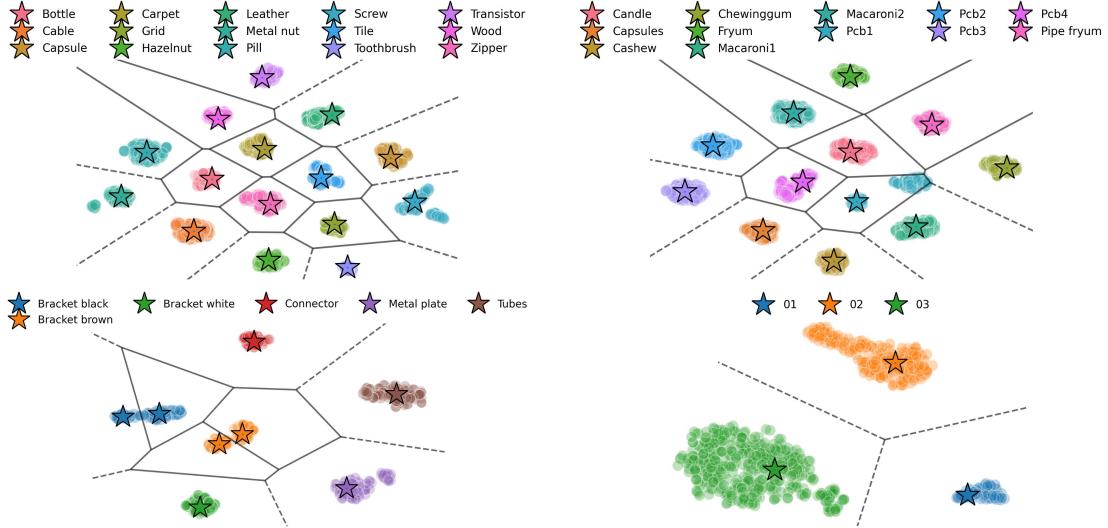


Figure 4: Visualization of semantic cluster and test samples. Unknown training setting. The star marks are the centroid of semantic clusters. The circles are test samples. The colors mean class for each cluster and sample.

multiple distinct semantic clusters. This indicates that even within a single class, normal data can be semantically diverse and may behave as different classes in the embedding space. Such diversity suggests that semantic representations may offer a more meaningful structure for anomaly detection than rigid class labels.

This observation aligns with our empirical findings. For example, in Table 4, HierCore occasionally achieves better performance under unknown evaluation conditions (i.e., without class labels) than under known ones. This supports the idea that semantic clustering may lead to more effective anomaly detection than label-based grouping. In contrast, Table 5 shows a slight performance drop when models trained with class labels are evaluated without them. This can be attributed to a misalignment between semantic clusters and class labels, suggesting that class labels may not always reflect semantic structure.

These findings collectively demonstrate that HierCore’s memory bank construction based on semantic clustering provides a more robust and generalizable foundation for anomaly detection than approaches strictly based on predefined class labels.

4.6. Memory Bank and Inference Time

HierCore constructs memory banks by clustering input data based on semantic similarity and assigning a separate memory bank to each cluster. In contrast, PatchCore builds a single global memory bank using all normal data, regardless of

Table 6: Comparison computation costs for PatchCore and HierCore. The computation time of HierCore for Memory Bank is the sum of the computation times of Memory Bank and FINCH.

Dataset	Memory Bank (minutes)		Evaluation (seconds)		Inference (FPS)	
	PatchCore	HierCore	PatchCore	HierCore	PatchCore	HierCore
MVTecAD	317	24 + 1	196	96	8.82	17.98
VisA	1,886	154 + 2	462	131	4.67	16.51
MPDD	17	3 + 0	29	25	15.89	18.09
BTAD	75	30 + 0	58	46	12.82	15.97

class or semantic grouping. As discussed in Section 3.4, this structural difference allows HierCore to significantly reduce the computational cost of memory bank construction and achieve more efficient nearest-neighbor search, enabling faster anomaly inference.

To quantify this efficiency, Table 6 compares three metrics across four industrial datasets: memory bank construction time (in hours), total inference time for the entire evaluation dataset (in seconds), and the number of frames processed per second (FPS) during inference.

On datasets with a large number of classes, such as MVTec AD, HierCore achieves approximately 13× faster memory bank construction compared to PatchCore. For datasets with a large volume of data, such as VisA, the difference is even more pronounced: while PatchCore takes nearly 30 hours to construct the memory bank, HierCore completes the process over 12 times faster. In terms of inference time, HierCore records an average of 3.5× speed-up compared to PatchCore. Similarly, HierCore shows a 3.5× improvement in FPS on average. Notably, PatchCore exhibits significant FPS variance across datasets, with up to a 3.4× difference between its fastest and slowest cases. In contrast, HierCore maintains stable performance across datasets, with a maximum FPS variation within 1.1×.

These results highlight that HierCore is not only faster but also more consistent in runtime efficiency, making it highly suitable for real-time anomaly detection in multi-class or large-scale industrial settings.

5. Conclusion

This study addressed a practical challenge in extending OC-UIAD methods to multi-class scenarios, going beyond mere performance improvement to focus on flexible applicability depending on the availability of class information during training

and evaluation. To this end, we defined two requirements for multi-class image anomaly detection and re-evaluated existing MC-UIAD approaches in terms of their ability to meet these criteria.

We proposed HierCore, a novel memory-based framework that can operate effectively regardless of whether class labels are available. Extensive experiments on four industrial benchmark datasets demonstrated that HierCore achieves robust and consistent performance across both known and unknown training and evaluation settings. While most previous MC-UIAD methods rely on reconstruction-based approaches, our study explored the scalability and robustness of memory bank-based methods. Notably, we found that PatchCore, although originally designed as an OC-UIAD method, retains strong performance even in multi-class environments.

HierCore addresses the main limitations of PatchCore by clustering normal data based on semantic features and constructing independent memory banks for each cluster. This hierarchical structure significantly reduces the computational burden and inference latency caused by scaling to large datasets. Our experiments showed that even when using only 10% of patch features to build the memory bank, HierCore maintained high anomaly detection accuracy.

We hope that this work encourages further research into MC-UIAD methods that satisfy both practical requirements proposed herein and are adaptable to diverse real-world industrial settings.

References

- [1] J. Liu, G. Xie, J. Wang, S. Li, C. Wang, F. Zheng, Y. Jin, Deep visual anomaly detection in industrial manufacturing: A survey, arXiv preprint arXiv:2301.11514 1 (2023).
- [2] T. Xiang, Y. Zhang, Y. Lu, A. L. Yuille, C. Zhang, W. Cai, Z. Zhou, Squid: Deep feature in-painting for unsupervised anomaly detection, in: CVPR, 2023, pp. 23890–23901.
- [3] T. Schlegl, P. Seeböck, S. M. Waldstein, G. Langs, U. Schmidt-Erfurth, f-anogan: Fast unsupervised anomaly detection with generative adversarial networks, Medical Image Analysis 54 (2019) 30–44.
- [4] W. Sultani, C. Chen, M. Shah, Real-world anomaly detection in surveillance videos, in: CVPR, 2018, pp. 6479–6488.
- [5] S. Chang, Y. Li, S. Shen, J. Feng, Z. Zhou, Contrastive attention for video anomaly detection, IEEE Trans. Multimedia 24 (2021) 4067–4076.

- [6] K. K. Santhosh, D. P. Dogra, P. P. Roy, Anomaly detection in road traffic using visual surveillance: A survey, *ACM Comput. Surv.* 53 (6) (2020) 1–26.
- [7] H. He, J. Zhang, H. Chen, X. Chen, Z. Li, X. Chen, Y. Wang, C. Wang, L. Xie, A diffusion-based framework for multi-class anomaly detection, in: AAAI, Vol. 38, 2024, pp. 8472–8480.
- [8] J. Zhang, X. Chen, Y. Wang, C. Wang, Y. Liu, X. Li, M.-H. Yang, D. Tao, Exploring plain vit features for multi-class unsupervised visual anomaly detection, *Comput. Vis. Image Underst.* (2025).
- [9] J. Zhang, C. Wang, X. Li, G. Tian, Z. Xue, Y. Liu, G. Pang, D. Tao, Learning feature inversion for multi-class anomaly detection under general-purpose coco-ad benchmark, arXiv preprint arXiv:2404.10760 (2024).
- [10] H. He, Y. Bai, J. Zhang, Q. He, H. Chen, Z. Gan, C. Wang, X. Li, G. Tian, L. Xie, Mambaad: Exploring state space models for multi-class unsupervised anomaly detection, in: NeurIPS, 2024.
- [11] X. Jiang, Y. Chen, Q. Nie, J. Liu, Y. Liu, C. Wang, F. Zheng, Toward multi-class anomaly detection: Exploring class-aware unified model against inter-class interference, arXiv preprint arXiv:2403.14213 (2024).
- [12] Z. You, L. Cui, Y. Shen, K. Yang, X. Lu, Y. Zheng, X. Le, A unified model for multi-class anomaly detection, in: NeurIPS, Vol. 35, 2022, pp. 4571–4584.
- [13] L. He, Z. Jiang, J. Peng, W. Zhu, L. Liu, Q. Du, X. Hu, M. Chi, Y. Wang, C. Wang, Learning unified reference representation for unsupervised multi-class anomaly detection, in: ECCV, 2024, pp. 216–232.
- [14] Y. Lee, H. Lim, S. Jang, H. Yoon, Uniformaly: Towards task-agnostic unified framework for visual anomaly detection, arXiv preprint arXiv:2307.12540 (2023).
- [15] Y. Zhou, X. Xu, Z. Sun, J. Song, A. Cichocki, H. T. Shen, Vq-flow: Tamming normalizing flows for multi-class anomaly detection via hierarchical vector quantization, arXiv preprint arXiv:2409.00942 (2024).
- [16] H. Song, M. Kim, D. Park, Y. Shin, J.-G. Lee, Learning from noisy labels with deep neural networks: A survey, *IEEE Trans. Neural Netw. Learn. Syst.* 34 (11) (2022) 8135–8153.

- [17] D. Gong, L. Liu, V. Le, B. Saha, M. R. Mansour, S. Venkatesh, A. v. d. Hengel, Memorizing normality to detect anomaly: Memory-augmented deep autoencoder for unsupervised anomaly detection, in: ICCV, 2019, pp. 1705–1714.
- [18] R. Lu, Y. Wu, L. Tian, D. Wang, B. Chen, X. Liu, R. Hu, Hierarchical vector quantized transformer for multi-class unsupervised anomaly detection, in: NeurIPS, Vol. 36, 2023, pp. 8487–8500.
- [19] S. Lee, S. Lee, B. C. Song, Cfa: Coupled-hypersphere-based feature adaptation for target-oriented anomaly localization, IEEE Access 10 (2022) 78446–78454.
- [20] T. Defard, A. Setkov, A. Loesch, R. Audigier, Padim: a patch distribution modeling framework for anomaly detection and localization, in: ICPR Workshops, 2021, pp. 475–489.
- [21] K. Roth, L. Pemula, J. Zepeda, B. Schölkopf, T. Brox, P. Gehler, Towards total recall in industrial anomaly detection, in: CVPR, 2022, pp. 14318–14328.
- [22] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, Imagenet: A large-scale hierarchical image database, in: CVPR, 2009, pp. 248–255.
- [23] M. D. Zeiler, R. Fergus, Visualizing and understanding convolutional networks, in: ECCV, 2014, pp. 818–833.
- [24] S. Sarfraz, V. Sharma, R. Stiefelhagen, Efficient parameter-free clustering using first neighbor relations, in: CVPR, 2019, pp. 8934–8943.
- [25] P. J. Rousseeuw, Silhouettes: a graphical aid to the interpretation and validation of cluster analysis, Journal of computational and applied mathematics 20 (1987) 53–65.
- [26] O. Sener, S. Savarese, Active learning for convolutional neural networks: A core-set approach, in: ICLR, 2018.
- [27] J. Zhang, H. He, Z. Gan, Q. He, Y. Cai, Z. Xue, Y. Wang, C. Wang, L. Xie, Y. Liu, Ader: A comprehensive benchmark for multi-class visual anomaly detection, arXiv preprint arXiv:2406.03262 (2024).
- [28] P. Bergmann, M. Fauser, D. Sattlegger, C. Steger, Mvtec ad—a comprehensive real-world dataset for unsupervised anomaly detection, in: CVPR, 2019, pp. 9592–9600.

- [29] Y. Zou, J. Jeong, L. Pemula, D. Zhang, O. Dabeer, Spot-the-difference self-supervised pre-training for anomaly detection and segmentation, in: ECCV, 2022, pp. 392–408.
- [30] S. Jezek, M. Jonak, R. Burget, P. Dvorak, M. Skotak, Deep learning-based defect detection of metal parts: evaluating current methods in complex conditions, in: Int. Congr. Ultra Modern Telecommun. Control Syst. Workshops, IEEE, 2021, pp. 66–71.
- [31] P. Mishra, R. Verk, D. Fornasier, C. Piciarelli, G. L. Foresti, Vt-adl: A vision transformer network for image anomaly detection and localization, in: IEEE Int. Symp. Ind. Electron., 2021, pp. 01–06.
- [32] V. Zavrtanik, M. Kristan, D. Skočaj, Draem-a discriminatively trained reconstruction embedding for surface anomaly detection, in: ICCV, 2021, pp. 8330–8339.
- [33] Z. Liu, Y. Zhou, Y. Xu, Z. Wang, Simplenet: A simple network for image anomaly detection and localization, in: CVPR, 2023, pp. 20402–20411.
- [34] X. Zhang, M. Xu, X. Zhou, Realnet: A feature selection network with realistic synthetic anomaly for anomaly detection, in: CVPR, 2024, pp. 16699–16708.
- [35] H. Deng, X. Li, Anomaly detection via reverse distillation from one-class embedding, in: CVPR, 2022, pp. 9737–9746.
- [36] Z. Gu, J. Zhang, L. Liu, X. Chen, J. Peng, Z. Gan, G. Jiang, A. Shu, Y. Wang, L. Ma, Rethinking reverse distillation for multi-modal anomaly detection, in: AAAI, Vol. 38, 2024, pp. 8445–8453.
- [37] X. Zhang, S. Li, X. Li, P. Huang, J. Shan, T. Chen, Destseg: Segmentation guided denoising student-teacher for anomaly detection, in: CVPR, 2023, pp. 3914–3923.
- [38] D. Gudovskiy, S. Ishizaka, K. Kozuka, Cflow-ad: Real-time unsupervised anomaly detection with localization via conditional normalizing flows, in: WACV, 2022, pp. 98–107.
- [39] J. Lei, X. Hu, Y. Wang, D. Liu, Pyramidflow: High-resolution defect contrastive localization using pyramid normalizing flow, in: CVPR, 2023, pp. 14143–14152.
- [40] S. Zagoruyko, N. Komodakis, Wide residual networks, in: BMVC, 2016.

- [41] M. Douze, A. Guzhva, C. Deng, J. Johnson, G. Szilvassy, P.-E. Mazaré, M. Lomeli, L. Hosseini, H. Jégou, The faiss library, arXiv preprint arXiv:2401.08281 (2024).
- [42] P. Bergmann, M. Fauser, D. Sattlegger, C. Steger, Uninformed students: Student-teacher anomaly detection with discriminative latent embeddings, in: CVPR, 2020, pp. 4183–4192.

Appendix A. Evaluation of Performance Gap between One-class and Multi-class Image Anomaly Detection

Table A.7: Anomaly detection performance on the MVTecAD dataset. The table presents a comparison between one-class and multi-class unsupervised image anomaly detection models. One-class models are evaluated under both one-class and multi-class settings. All experiments are conducted assuming unknown classes during training and known classes during evaluation.

Model	Image-level					Pixel-level					
	mAUROC	mAP	mF1-max	mAD	mAUROC	mAP	mF1-max	mAUPRO	mIoU-max	mAD	
One-class	DRAEM	0.716	0.545	0.864	0.763	0.868	0.836	0.816	0.715	0.567	0.476
	SimpleNet	0.981	0.954	0.993	0.983	0.976	0.957	0.983	0.965	0.973	0.968
	RealNet	0.966	0.848	0.986	0.941	0.959	0.909	0.970	0.899	0.932	0.726
	RD	0.985	0.936	0.993	0.972	0.976	0.956	0.985	0.955	0.974	0.958
	RD++	0.986	0.979	0.994	0.988	0.980	0.964	0.987	0.977	0.973	0.955
	DesTSeg	0.964	0.964	0.984	0.986	0.961	0.962	0.970	0.971	0.932	0.920
	CFLOW-AD	0.950	0.916	0.981	0.967	0.953	0.934	0.961	0.939	0.967	0.957
	PyramidFlow	0.878	0.702	0.944	0.855	0.904	0.855	0.909	0.804	0.946	0.800
	CFA	0.921	0.576	0.966	0.783	0.950	0.847	0.946	0.735	0.954	0.548
	PatchCore	0.993	0.992	0.998	0.998	0.986	0.985	0.992	0.992	0.981	0.980
Multi-class	UniAD	-	0.924	-	0.973	-	0.952	-	0.950	-	0.429
	DiAD	-	0.889	-	0.958	-	0.935	-	0.927	-	0.893
	VITAD	-	0.980	-	0.991	-	0.968	-	0.980	-	0.977
	InvAD	-	0.981	-	0.991	-	0.976	-	0.983	-	0.981
	InvAD-lite	-	0.979	-	<u>0.992</u>	-	0.968	-	0.980	-	0.979
	MambaAD	-	0.974	-	0.991	-	0.969	-	0.978	-	0.974
	HierCore (Ours)	-	0.992	-	0.998	-	0.986	-	0.992	-	0.981

Table A.8: Anomaly detection performance on the VisA dataset. The table presents a comparison between one-class and multi-class unsupervised image anomaly detection models. One-class models are evaluated under both one-class and multi-class settings. All experiments are conducted assuming unknown classes during training and known classes during evaluation.

Model	Image-level					Pixel-level															
	mAUROC	mAP	mF1-max	mAD	mAUROC	mAP	mF1-max	mAURO	mIoU-max	mAD											
One-class	DRAEM	0.727	0.551	0.744	0.624	0.759	0.729	0.743	0.635	0.488	0.375	0.012	0.006	0.036	0.017	0.246	0.100	0.019	0.009	0.160	0.101
	SimpleNet	0.945	0.864	0.951	0.891	0.910	0.828	0.936	0.861	0.980	0.966	0.350	0.340	0.395	0.378	0.866	0.792	0.268	0.257	0.572	0.547
	RealNet	0.922	0.714	0.941	0.795	0.888	0.747	0.917	0.752	0.864	0.610	0.410	0.257	0.454	0.226	0.661	0.274	0.313	0.135	0.540	0.300
	RD	0.959	0.906	0.962	0.909	0.925	0.893	0.949	0.903	0.987	0.980	0.409	0.354	0.453	0.425	0.934	0.919	0.306	0.279	0.618	0.591
	RD++	0.952	0.939	0.958	0.947	0.917	0.902	0.943	0.929	0.988	0.984	0.406	0.423	0.450	0.463	0.930	0.919	0.305	0.312	0.616	0.620
	DesTSeg	0.913	0.899	0.923	0.914	0.876	0.867	0.904	0.893	0.903	0.867	0.367	0.466	0.417	0.472	0.703	0.611	0.270	0.327	0.534	0.549
	CFLOW-AD	0.901	0.865	0.910	0.888	0.863	0.849	0.892	0.867	0.982	0.977	0.329	0.339	0.386	0.372	0.884	0.868	0.252	0.249	0.567	0.561
	PyramidFlow	0.902	0.582	0.908	0.663	0.867	0.744	0.892	0.663	0.952	0.770	0.243	0.072	0.312	0.096	0.808	0.428	0.191	0.056	0.501	0.284
	CFA	0.861	0.663	0.897	0.743	0.847	0.742	0.868	0.716	0.956	0.813	0.290	0.221	0.341	0.262	0.770	0.508	0.223	0.170	0.516	0.395
	PatchCore	0.963	0.961	0.968	0.967	0.927	0.930	0.953	0.953	0.981	0.980	0.505	0.514	0.512	0.517	0.909	0.907	0.356	0.361	0.653	0.656
Multi-class	UniAD	-	0.888	-	0.908	-	0.858	-	0.884	-	0.983	-	0.337	-	0.390	-	0.855	-	0.257	-	0.564
	DiAD	-	0.848	-	0.885	-	0.869	-	0.867	-	0.825	-	0.179	-	0.232	-	0.445	-	0.149	-	0.366
	VITAD	-	0.905	-	0.917	-	0.863	-	0.895	-	0.982	-	0.366	-	0.411	-	0.851	-	0.276	-	0.577
	InvAD	-	0.955	-	0.958	-	0.920	-	0.944	-	0.989	-	0.431	-	0.470	-	0.926	-	0.327	-	0.628
	InvAD-lite	-	0.949	-	0.952	-	0.907	-	0.936	-	0.986	-	0.402	-	0.440	-	0.931	-	0.298	-	0.611
	MambaAD	-	0.941	-	0.948	-	0.893	-	0.927	-	0.985	-	0.395	-	0.437	-	0.914	-	0.295	-	0.605
	HierCore (Ours)	-	0.963	-	0.968	-	0.927	-	0.953	-	0.981	-	0.498	-	0.507	-	0.909	-	0.350	-	0.649

Table A.9: Anomaly detection performance on the MPDD dataset. The table presents a comparison between one-class and multi-class unsupervised image anomaly detection models. One-class models are evaluated under both one-class and multi-class settings. All experiments are conducted assuming unknown classes during training and known classes during evaluation.

Model	Image-level					Pixel-level															
	mAUROC	mAP	mF1-max	mAD	mAUROC	mAP	mF1-max	mAURO	mIoU-max	mAD											
One-class	DRAEM	0.706	0.356	0.785	0.533	0.842	0.725	0.778	0.538	0.427	0.421	0.019	0.022	0.053	0.046	0.242	0.190	0.029	0.025	0.154	0.141
	SimpleNet	0.954	0.884	0.957	0.920	0.940	0.879	0.950	0.894	0.938	0.965	0.314	0.320	0.340	0.346	0.788	0.890	0.240	0.245	0.524	0.553
	RealNet	0.764	0.851	0.810	0.902	0.803	0.883	0.792	0.879	0.876	0.833	0.229	0.361	0.267	0.396	0.731	0.681	0.186	0.282	0.458	0.511
	RD	0.944	0.913	0.965	0.936	0.922	0.918	0.944	0.922	0.977	0.983	0.432	0.404	0.461	0.418	0.936	0.955	0.343	0.314	0.630	0.615
	RD++	0.918	0.902	0.932	0.933	0.914	0.905	0.921	0.913	0.979	0.985	0.409	0.430	0.429	0.441	0.940	0.955	0.314	0.336	0.614	0.629
	DesTSeg	0.918	0.913	0.938	0.908	0.907	0.902	0.921	0.908	0.886	0.820	0.363	0.326	0.380	0.346	0.734	0.633	0.284	0.256	0.529	0.476
	CFLOW-AD	0.850	0.757	0.880	0.801	0.865	0.817	0.865	0.792	0.977	0.968	0.339	0.263	0.341	0.280	0.913	0.895	0.248	0.201	0.564	0.521
	PyramidFlow	0.843	0.736	0.865	0.770	0.843	0.794	0.850	0.767	0.973	0.941	0.291	0.211	0.311	0.178	0.883	0.772	0.218	0.104	0.535	0.441
	CFA	0.866	0.816	0.872	0.877	0.876	0.857	0.872	0.850	0.930	0.849	0.196	0.196	0.217	0.229	0.701	0.535	0.161	0.166	0.441	0.395
	PatchCore	0.945	0.948	0.968	0.969	0.933	0.938	0.949	0.952	0.985	0.987	0.479	0.484	0.501	0.504	0.945	0.950	0.368	0.371	0.656	0.659
Multi-class	UniAD	-	0.707	-	0.754	-	0.794	-	0.752	-	0.942	-	0.139	-	0.203	-	0.799	-	0.128	-	0.442
	DiAD	-	0.683	-	0.779	-	0.801	-	0.754	-	0.904	-	0.109	-	0.131	-	0.661	-	0.082	-	0.377
	VITAD	-	0.870	-	0.901	-	0.872	-	0.881	-	0.977	-	0.354	-	0.375	-	0.926	-	0.278	-	0.582
	InvAD	-	0.931	-	0.943	-	0.928	-	0.934	-	0.983	-	0.428	-	0.460	-	0.947	-	0.341	-	0.632
	InvAD-lite	-	0.909	-	0.929	-	0.895	-	0.911	-	0.980	-	0.397	-	0.426	-	0.940	-	0.309	-	0.610
	MambaAD	-	0.877	-	0.927	-	0.905	-	0.903	-	0.976	-	0.349	-	0.394	-	0.926	-	0.278	-	0.585
	HierCore (Ours)	-	0.943	-	0.967	-	0.934	-	0.948	-	0.985	-	0.477	-	0.500	-	0.945	-	0.368	-	0.655

Table A.10: Anomaly detection performance on the BTAD dataset. The table presents a comparison between one-class and multi-class unsupervised image anomaly detection models. One-class models are evaluated under both one-class and multi-class settings. All experiments are conducted assuming unknown classes during training and known classes during evaluation.

Model		Image-level					Pixel-level														
		mAUROC	mAP	mf1-max	mAD	mAUROC	mAP	mf1-max	mAURO	miIoU-max	mAD										
One-class	DRAEM	0.717	0.713	0.762	0.785	0.781	0.780	0.754	0.759	0.502	0.490	0.110	0.037	0.086	0.065	0.169	0.162	0.045	0.034	0.182	0.158
	SimpleNet	0.936	0.932	0.978	0.973	0.942	0.933	0.952	0.946	0.965	0.963	0.458	0.415	0.471	0.443	0.704	0.698	0.310	0.286	0.582	0.561
	RealNet	0.950	0.897	0.969	0.953	0.928	0.928	0.949	0.926	0.954	0.840	0.538	0.481	0.570	0.527	0.750	0.534	0.401	0.366	0.643	0.550
	RD	0.943	0.944	0.952	0.966	0.928	0.940	0.941	0.950	0.980	0.981	0.583	0.596	0.583	0.592	0.801	0.807	0.413	0.421	0.672	0.679
	RD++	0.947	0.946	0.969	0.978	0.936	0.941	0.951	0.955	0.980	0.980	0.580	0.596	0.588	0.598	0.795	0.790	0.418	0.428	0.672	0.678
	DesTSeg	0.885	0.928	0.941	0.959	0.918	0.923	0.915	0.937	0.944	0.922	0.376	0.348	0.419	0.443	0.728	0.700	0.268	0.290	0.547	0.541
	CFLOW-AD	0.934	0.912	0.972	0.948	0.935	0.883	0.947	0.914	0.970	0.968	0.492	0.456	0.418	0.501	0.749	0.727	0.278	0.338	0.581	0.598
	PyramidFlow	0.864	0.870	0.756	0.831	0.772	0.810	0.797	0.837	0.927	0.909	0.395	0.296	0.351	0.269	0.694	0.641	0.249	0.183	0.523	0.460
Multi-class	CFA	0.931	0.927	0.981	0.975	0.955	0.935	0.955	0.946	0.972	0.963	0.569	0.474	0.562	0.502	0.740	0.695	0.391	0.336	0.648	0.594
	PatchCore	0.941	0.944	0.978	0.978	0.938	0.947	0.952	0.956	0.972	0.580	0.582	0.563	0.564	0.753	0.753	0.393	0.394	0.652	0.653	
		-	0.949	-	0.983	-	0.948	-	0.960	-	0.972	-	0.501	-	0.537	-	0.779	-	0.368	-	0.631
		-	0.901	-	0.884	-	0.926	-	0.904	-	0.917	-	0.196	-	0.267	-	0.704	-	0.157	-	0.448
		-	0.940	-	0.972	-	0.938	-	0.950	-	0.975	-	0.639	-	0.600	-	0.711	-	0.401	-	0.665
		-	0.959	-	0.975	-	0.950	-	0.961	-	0.981	-	0.621	-	0.612	-	0.799	-	0.441	-	0.691
		-	0.931	-	0.974	-	0.946	-	0.950	-	0.979	-	0.592	-	0.597	-	0.786	-	0.426	-	0.676
		-	0.921	-	0.969	-	0.938	-	0.942	-	0.976	-	0.521	-	0.556	-	0.777	-	0.386	-	0.643
		-	0.941	-	0.978	-	0.938	-	0.952	-	0.972	-	0.580	-	0.563	-	0.753	-	0.393	-	0.652

Appendix B. Evaluation of Requirements for Multi-Class Unsupervised Image Anomaly Detection

Table B.11: Image- and pixel-level anomaly detection performance on the MVTecAD dataset based on F1-score with optimal thresholding. Models are trained under an unknown classes setting. During evaluation, F1-scores for known classes are computed using class-specific optimal thresholds, while a single optimal threshold is used for unknown-class evaluation. Diff. Ratio indicates the ratio of F1-scores under the unknown-class evaluation to those under the known-class evaluation.

Categories	Evaluation	Image-level						Pixel-level					
		UniAD	ViTAD	InvAD	MambaAD	PatchCore	HierCore	UniAD	ViTAD	InvAD	MambaAD	PatchCore	HierCore
Bottle	Known	94.6	98.4	99.2	99.2	99.2	99.2	71.0	76.5	73.7	76.5	79.4	79.5
	Unknown	94.7	99.2	100.0	100.0	99.2	99.2	68.5	74.1	73.6	76.4	79.4	79.5
Cable	Known	88.1	93.0	94.6	95.7	97.3	96.3	51.2	47.5	53.1	47.9	57.5	64.7
	Unknown	80.3	78.0	76.0	81.1	76.0	96.3	47.6	45.1	52.9	45.3	52.3	65.4
Capsule	Known	90.4	94.3	95.5	93.4	97.3	97.7	36.4	48.3	50.7	44.7	56.5	56.0
	Unknown	86.7	84.0	80.2	58.4	84.0	97.7	35.8	38.8	41.1	37.9	45.9	56.0
Hazelnut	Known	93.7	98.6	99.3	98.6	99.3	99.3	56.2	65.3	61.4	66.0	68.2	68.2
	Unknown	81.4	77.8	85.9	88.6	78.7	99.3	50.7	51.2	55.6	54.4	57.7	68.2
Metal nut	Known	96.3	97.9	99.5	98.4	98.9	98.9	67.0	77.6	81.4	80.0	87.0	87.1
	Unknown	96.7	93.9	96.4	97.9	93.5	98.9	55.4	67.8	80.3	75.4	87.0	87.1
Pill	Known	91.2	96.0	97.1	94.4	96.1	96.5	26.7	75.4	69.1	61.0	74.4	74.3
	Unknown	91.6	96.0	96.3	94.3	96.0	96.5	18.0	61.3	43.0	52.7	64.6	74.3
Screw	Known	89.3	91.1	95.1	91.5	96.7	97.1	29.4	39.2	47.4	45.6	52.1	52.0
	Unknown	87.3	89.8	80.2	71.0	77.3	97.1	29.2	34.8	38.6	43.9	38.1	52.0
Toothbrush	Known	84.1	95.1	93.5	93.5	96.6	96.6	50.6	61.0	59.6	60.2	66.0	65.4
	Unknown	85.3	92.3	95.2	95.2	96.7	96.6	49.1	52.7	51.6	53.6	63.9	65.4
Transistor	Known	82.5	86.0	91.6	94.7	98.7	98.7	68.0	55.1	63.2	60.7	61.1	61.3
	Unknown	64.0	69.6	72.1	80.0	88.9	98.7	68.0	48.1	58.0	59.2	53.8	61.3
Zipper	Known	91.3	96.6	98.3	96.7	98.3	98.3	41.4	51.0	57.0	58.8	71.4	71.7
	Unknown	85.6	82.5	93.8	86.9	94.2	98.3	31.1	24.7	32.7	39.2	40.4	71.7
Carpet	Known	97.7	98.9	96.5	98.9	96.0	96.0	55.7	64.5	60.9	64.8	67.8	67.8
	Unknown	94.2	99.4	95.6	99.4	96.0	96.0	52.7	62.5	59.9	62.9	65.9	67.8
Grid	Known	93.8	98.2	96.6	99.1	98.2	99.1	34.1	37.0	46.2	47.9	53.9	54.5
	Unknown	89.1	94.4	96.5	94.4	99.1	99.1	25.0	35.2	46.1	47.9	52.8	54.5
Leather	Known	99.5	99.5	99.5	99.5	99.5	99.5	43.7	57.4	52.2	49.1	55.8	55.4
	Unknown	100.0	100.0	98.9	100.0	100.0	99.5	31.3	33.3	36.1	31.7	51.4	55.4
Tile	Known	90.7	99.4	99.4	93.8	98.2	98.2	47.6	69.1	61.8	52.1	70.9	71.0
	Unknown	83.6	88.4	88.9	85.3	98.8	98.2	47.5	56.9	59.9	50.4	69.5	71.0
Wood	Known	94.9	96.6	97.5	95.9	96.7	96.7	45.6	58.6	52.1	48.7	58.5	58.3
	Unknown	89.6	92.3	94.5	93.8	96.7	96.7	45.4	49.1	47.0	41.2	58.1	58.3
Average	Known	91.9	96.0	96.9	96.2	97.8	97.9	48.3	58.9	59.3	57.6	65.4	65.8
	Unknown	87.3	89.2	90.0	88.4	91.7	97.9	43.7	49.0	51.8	51.5	58.7	65.9
	Diff. Ratio	95.1%	92.9%	92.9%	91.9%	93.8%	100.0%	90.4%	83.2%	87.2%	89.4%	89.8%	100.1%

Table B.12: Image- and pixel-level anomaly detection performance on the VisA dataset based on F1-score with optimal thresholding. Models are trained under an unknown classes setting. During evaluation, F1-scores for known classes are computed using class-specific optimal thresholds, while a single optimal threshold is used for unknown-class evaluation. Diff. Ratio indicates the ratio of F1-scores under the unknown-class evaluation to those under the known-class evaluation.

Categories	Evaluation	Image-level						Pixel-level					
		UniAD	ViTAD	InvAD	MambaAD	PatchCore	HierCore	UniAD	ViTAD	InvAD	MambaAD	PatchCore	HierCore
PCB1	Known	90.2	91.2	92.6	89.9	97.0	96.5	59.6	62.1	76.4	71.3	81.1	75.9
	Unknown	88.0	84.4	91.1	90.4	96.6	96.5	56.0	61.9	70.4	71.2	76.5	76.6
PCB2	Known	82.9	84.3	92.3	86.3	93.5	93.0	17.0	21.2	24.0	22.6	37.7	36.1
	Unknown	80.3	81.2	90.0	85.7	93.6	93.0	16.9	21.0	23.5	22.6	32.6	35.5
PCB3	Known	78.3	83.5	91.8	86.4	93.5	92.9	24.1	26.4	28.4	26.8	43.9	44.0
	Unknown	76.7	83.6	90.6	87.0	89.1	92.9	20.3	26.0	27.7	26.6	39.0	44.0
PCB4	Known	93.1	95.6	97.4	96.4	97.5	97.5	33.6	48.3	45.2	46.5	49.1	49.1
	Unknown	88.4	89.7	95.2	94.8	95.2	97.5	28.4	48.3	44.7	46.5	48.9	49.1
Macaroni1	Known	75.1	74.9	87.4	80.6	95.1	94.2	16.2	19.4	29.8	24.7	39.6	39.1
	Unknown	67.1	65.7	70.4	69.0	88.8	94.2	16.2	12.4	28.5	22.9	36.0	39.1
Macaroni2	Known	67.0	74.0	80.0	76.8	71.5	70.3	12.4	10.4	18.7	17.8	32.2	32.1
	Unknown	64.2	67.4	69.1	71.9	71.2	70.3	11.0	2.6	17.6	17.1	31.9	32.1
Capsules	Known	77.0	79.8	87.7	88.6	82.1	81.6	44.8	41.4	65.2	60.3	66.5	67.7
	Unknown	77.5	76.9	87.2	87.3	73.5	81.6	41.8	39.4	59.0	51.2	50.3	67.7
Candles	Known	87.9	85.0	90.1	91.5	95.9	96.5	32.8	26.6	34.5	31.0	43.2	43.6
	Unknown	87.3	79.6	77.7	90.7	88.9	96.5	32.5	16.1	29.1	30.8	38.2	43.6
Cashew	Known	91.0	85.2	94.4	88.8	95.5	95.5	59.4	62.9	61.5	53.0	61.4	61.2
	Unknown	84.6	84.5	89.0	89.3	94.5	95.5	57.2	57.8	55.3	49.3	54.6	61.2
Chewing gum	Image	94.9	92.6	95.3	94.2	97.5	97.5	58.4	59.0	62.6	59.8	58.8	51.1
	Pixel	89.4	79.0	91.2	79.0	98.0	97.5	44.0	32.5	49.1	42.0	44.0	51.1
Fryum	Image	84.7	90.2	91.8	88.7	91.9	92.1	53.1	51.1	53.1	52.6	48.7	48.6
	Pixel	83.4	90.2	88.5	85.9	91.7	92.1	41.6	28.3	37.8	35.5	36.7	48.6
Pipe fryum	Image	93.4	94.9	97.5	96.1	99.0	98.5	59.7	66.7	66.4	59.2	61.9	61.9
	Pixel	90.7	89.2	95.8	94.4	98.5	98.5	58.2	66.5	65.0	56.8	58.7	61.9
Average	Known	84.6	85.9	91.5	88.7	92.5	92.2	39.3	41.3	47.2	43.8	52.0	50.9
	Unknown	81.5	81.0	86.3	85.5	90.0	92.2	35.3	34.4	42.3	39.4	45.6	50.9
Diff. Ratio		96.3%	94.2%	94.3%	96.4%	97.2%	100.0%	90.0%	83.3%	89.7%	90.0%	87.8%	100.0%

Table B.13: Image- and pixel-level anomaly detection performance on the MPDD dataset based on F1-score with optimal thresholding. Models are trained under an unknown classes setting. During evaluation, F1-scores for known classes are computed using class-specific optimal thresholds, while a single optimal threshold is used for unknown-class evaluation. Diff. Ratio indicates the ratio of F1-scores under the unknown-class evaluation to those under the known-class evaluation.

Categories	Evaluation	Image-level						Pixel-level					
		UniAD	ViTAD	InvAD	MambaAD	PatchCore	HierCore	UniAD	ViTAD	InvAD	MambaAD	PatchCore	HierCore
Bracket Black	Known	78.9	80.4	80.4	80.0	84.1	83.9	1.6	9.6	9.6	16.8	28.7	28.4
	Unknown	78.5	81.4	81.4	78.6	76.5	83.5	1.4	0.0	0.0	13.9	7.5	30.2
Bracket Brown	Known	92.6	90.1	90.1	93.5	95.1	94.3	32.1	23.4	23.4	25.5	32.6	31.3
	Unknown	92.6	90.1	90.1	85.1	88.4	94.2	17.0	0.0	0.0	15.4	4.7	31.0
Bracket White	Known	71.8	73.3	73.3	87.7	86.8	85.2	1.1	2.2	2.2	22.6	25.7	26.3
	Unknown	67.4	66.7	66.7	88.1	86.8	85.2	0.8	0.0	0.0	17.5	0.6	26.3
Connector	Image	63.4	82.8	82.8	92.9	96.3	96.3	15.5	48.5	48.5	55.8	63.4	62.1
	Pixel	57.1	58.3	58.3	84.8	82.4	96.3	10.9	3.3	3.3	53.1	6.2	62.1
Metal Plate	Image	83.8	99.3	99.3	99.3	99.3	99.3	57.7	88.0	88.0	87.2	86.9	86.9
	Pixel	84.5	84.5	84.5	86.1	87.7	99.3	57.2	88.0	88.0	85.6	86.8	86.9
Tubes	Image	82.4	83.5	83.5	93.9	91.6	91.7	14.8	54.5	54.5	69.4	66.5	66.8
	Pixel	81.2	81.2	81.2	81.2	81.2	91.7	5.2	43.6	43.6	57.3	66.1	66.8
Average	Known	78.8	84.9	84.9	91.2	92.2	<u>91.8</u>	20.5	37.7	37.7	46.2	50.6	<u>50.3</u>
	Unknown	76.9	77.0	77.0	<u>84.0</u>	83.8	91.7	15.4	22.5	22.5	<u>40.5</u>	28.6	50.5
Diff. Ratio		97.5%	90.8%	90.8%	92.1%	90.9%	99.9%	75.4%	59.6%	59.6%	<u>87.5%</u>	56.5%	100.5%

Table B.14: Image- and pixel-level anomaly detection performance on the BTAD dataset based on F1-score with optimal thresholding. Models are trained under an unknown classes setting. During evaluation, F1-scores for known classes are computed using class-specific optimal thresholds, while a single optimal threshold is used for unknown-class evaluation. Diff. Ratio indicates the ratio of F1-scores under the unknown-class evaluation to those under the known-class evaluation.

Categories	Evaluation	Image-level						Pixel-level					
		UniAD	ViTAD	InvAD	MambaAD	PatchCore	HierCore	UniAD	ViTAD	InvAD	MambaAD	PatchCore	HierCore
01	Image	96.8	94.8	97.9	93.6	97.0	96.9	58.5	55.7	62.2	59.7	59.3	59.6
	Pixel	93.5	89.9	96.8	94.6	86.0	96.9	56.9	0.4	46.5	41.4	55.9	59.6
02	Image	92.7	93.1	93.9	92.5	93.2	93.0	51.8	67.6	62.4	58.3	60.3	60.2
	Pixel	83.4	90.4	79.5	82.2	64.2	93.0	51.8	67.4	62.1	58.0	60.2	60.2
03	Image	89.7	90.6	90.0	92.1	90.9	88.5	52.1	47.5	59.5	49.0	49.8	49.7
	Pixel	66.0	53.4	64.6	50.4	84.9	88.5	50.5	0.4	51.2	48.6	49.4	49.7
Average	Known	93.1	92.9	93.9	92.7	<u>93.7</u>	92.8	54.1	56.9	61.4	55.7	56.5	56.5
	Unknown	<u>81.0</u>	77.9	80.3	75.7	78.4	92.8	53.0	22.8	53.3	49.3	<u>55.2</u>	56.5
Diff. Ratio		87.0%	83.9%	85.5%	81.7%	83.7%	100.0%	98.0%	40.0%	86.8%	88.6%	97.7%	100.0%