

# A Comprehensive Library for Benchmarking Multi-class Visual Anomaly Detection

Jiangning Zhang<sup>1,2\*</sup> Haoyang He<sup>2\*</sup> Zhenye Gan<sup>1†</sup> Qingdong He<sup>1</sup> Yuxuan Cai<sup>3</sup>  
Zhucun Xue<sup>2</sup> Yabiao Wang<sup>1</sup> Chengjie Wang<sup>1</sup> Lei Xie<sup>2</sup> Yong Liu<sup>2‡</sup>  
<sup>1</sup>YouTu Lab, Tencent <sup>2</sup>Zhejiang University <sup>3</sup>Huazhong University of Science and Technology

Code: <https://github.com/zhangzjn/ADer>

## Abstract

*Visual anomaly detection aims to identify anomalous regions in images through unsupervised learning paradigms, with increasing application demand and value in fields such as industrial inspection and medical lesion detection. Despite significant progress in recent years, there is a lack of comprehensive benchmarks to adequately evaluate the performance of various mainstream methods across different datasets under the practical multi-class setting. The absence of standardized experimental setups can lead to potential biases in training epochs, resolution, and metric results, resulting in erroneous conclusions. This paper addresses this issue by proposing a comprehensive visual anomaly detection benchmark, **ADer**, which is a modular framework that is highly extensible for new methods. The benchmark includes multiple datasets from industrial and medical domains, implementing fifteen state-of-the-art methods and nine comprehensive metrics. Additionally, we have proposed the GPU-assisted **ADEval** package to address the slow evaluation problem of metrics like time-consuming mAU-PRO on large-scale data, significantly reducing evaluation time by more than 1000-fold. Through extensive experimental results, we objectively reveal the strengths and weaknesses of different methods and provide insights into the challenges and future directions of multi-class visual anomaly detection. We hope that **ADer** will become a valuable resource for researchers and practitioners in the field, promoting the development of more robust and generalizable anomaly detection systems.*

## 1. Introduction

In recent years, with the rapid advancement in model iteration and computational power, Visual Anomaly Detec-

tion (VAD) has made significant progress across various fields [7, 25]. It has become a crucial component in key tasks such as industrial quality inspection and medical lesion detection. Due to its unsupervised experimental setup, VAD demonstrates immense application value in real-world scenarios where the yield rate is high, defect samples are difficult to obtain, and potential defect patterns are diverse. However, the field faces challenges such as small dataset sizes and insufficient evaluation metrics, resulting in potentially unfair comparison outcomes due to differing training recipes among methods. Moreover, most methods have not been compared on the latest large-scale datasets (e.g., Real-IAD [32] and COCO-AD [42]) and new evaluation metrics (e.g., mAD [40] and mIoU-max [42]). The fundamental issue is the absence of standardized training strategies, akin to those in object detection, to evaluate different algorithms. Training epoch and resolution factors can potentially affect evaluation results, leading to erroneous conclusions.

To address this pressing issue, we believe that establishing a comprehensive and fair benchmark is crucial for the sustained and healthy development of this field. Therefore, we have constructed an integrated **ADer** library, benchmarking state-of-the-art methods by utilizing a unified evaluation interface under the more practical multi-class setting. This library is designed as a highly extensible modular framework (see Sec. 3), allowing for the easy implementation of new methods. Specifically, the framework integrates multiple datasets from industrial, medical, and general-purpose domains (see Sec. 3.2), and implements fifteen state-of-the-art methods (including augmentation-based, embedding-based, reconstruction-based, and hybrid methods, see Sec. 3.1) and nine comprehensive evaluation metrics (see Sec. 3.3), ensuring thorough and unbiased performance evaluation for each method. Additionally, to address the efficiency issue of evaluating time-consuming metrics like mAU-PRO on large-scale data, we have developed and open-sourced the GPU-assisted **ADEval** package (see Sec. 3.6), significantly reducing evaluation time by over 1000 times, making previously impractical extensive

\*Equal contribution.

†Project lead.

‡Corresponding author.

detailed evaluations feasible on large-scale datasets.

Through extensive and fair experiments, we objectively reveal the strengths and weaknesses of different visual anomaly detection methods, comparing their efficiency (*i.e.*, model parameter count and FLOPs) and training resource consumption across different datasets, as shown in Fig. 1. Detailed results and analyses (see Sec. 4 and Appendix) elucidate the challenges of multi-class visual anomaly detection and provide valuable insights for future research directions.

In summary, the contributions of this paper are as follows: **1) Comprehensive benchmark:** We introduce a modular and extensible library termed *ADer* for visual anomaly detection, which implements and evaluates 15 state-of-the-art anomaly detection methods on 11 popular datasets with 9 comprehensive evaluation metrics. **2) GPU-assisted evaluation package:** We develop and will open-source the *ADEval* package for large-scale evaluation, significantly reducing the evaluation time of complex metrics by over 1000 times. **3) Extensive experimental analysis:** We conduct extensive experiments to objectively evaluate the performance of different methods, providing insights into their strengths, weaknesses, and potential areas for improvement. **4) Open-source resources:** We will open-source the complete *ADer* code, making it a valuable resource for the research community and promoting further advancements in the field.

## 2. Background and Related Work

### 2.1. Problem Definition and Objective

Visual anomaly detection (VAD) is a critical task in computer vision, aiming at identifying patterns or instances in visual data that deviate significantly from the norm. These anomalies can manifest as industrial defects, medical lesion, or rare objects that are not typically present in the training data. The primary objective of VAD is to develop algorithms capable of discerning these irregularities with high accuracy and reliability. This task is particularly challenging due to the inherent variability and complexity of visual data, the scarcity of anomalous examples, and the need for robust generalization across diverse scenarios. In a formal context, multi-class VAD can be defined as follows: Given a training dataset  $D_{train} = \{x_1, x_2, \dots, x_n\}$  with  $C$  categories and each visual image  $x_i$  belonging to a specific category, the goal is to learn a unified AD model  $M$  that can predict an anomaly score  $s_i = M(x_i)$  for each image. This score reflects the likelihood of each pixel in  $x_i$  being an anomaly. The model  $M$  is typically trained on  $D_{train}$  that predominantly contains normal instances, with the assumption that anomalies are rare and not well-represented in the training set. Inevitably, there are some mislabeled or inaccurately labeled noisy samples, which constitute inherent biases within the dataset. These are typically disregarded

under standard settings. The performance of the model is then evaluated based on its ability to correctly identify anomalous images and their defect regions in a unified test-set that contains normal and anomalous images.

### 2.2. Challenges in Multi-class VAD

The complexity of VAD arises from several factors: **1) Data Imbalance:** Anomalies are rare, leading to highly imbalanced datasets where normal instances and region areas vastly outnumber anomalous ones. **2) Variability of Anomalies:** Anomalies can vary widely in appearance, making it difficult to capture all possible variations during training. **3) Context Sensitivity:** The definition of what constitutes an anomaly can be context-dependent, requiring models to understand the broader context in which the visual data is situated. **4) Efficiency Requirements:** Many applications of VAD require real-time processing and limited GPU memory, necessitating efficient and scalable algorithms. **5) Comprehensive and Fair Evaluation:** Current methods exhibit significant differences in training hyperparameters and insufficient evaluation of performance metrics, so it is necessary to benchmark them using fair and standardized criteria. In this benchmark study, we systematically evaluate a range of state-of-the-art VAD methods (Sec. 3.1) across multiple datasets (Sec. 3.2) and comprehensive metrics (Sec. 3.3). Our goal is to provide a comprehensive assessment of current capabilities, identify key challenges, and suggest directions for future research in visual anomaly detection.

### 2.3. Visual Anomaly Detection

Visual anomaly detection methods can generally be categorized into three types: **1) Augmentation-based methods** generate pseudo-supervised information for anomalies by creating abnormal regions [23, 37], constructing anomalous data [19, 38], or adding feature perturbations [27, 31]. This enables the model to learn the differences between normal and abnormal distributions. **2) Embedding-based methods** leverage pretrained models to extract powerful feature representations and judge anomalies in high-dimensional space. Typical approaches include distribution-map-based methods [14, 22], teacher-student frameworks [4, 33], and memory-bank techniques [13, 29]. **3) Reconstruction-based methods** use encoder-decoder architectures to locate anomalies by analyzing the reconstruction error. They typically include both image-level [2, 16, 24] and feature-level approaches [10, 40, 42]. There are also some hybrid methods [31, 35, 44] that attempt to integrate multiple techniques to further enhance model performance.

**Basic network structures of VAD.** Early visual anomaly detection methods typically employ UNet-based autoencoder architectures [1, 24, 37]. With advancements in foundational visual model structures [17, 26, 34, 39, 41, 43] and pretraining techniques [8, 18], more recent methods often

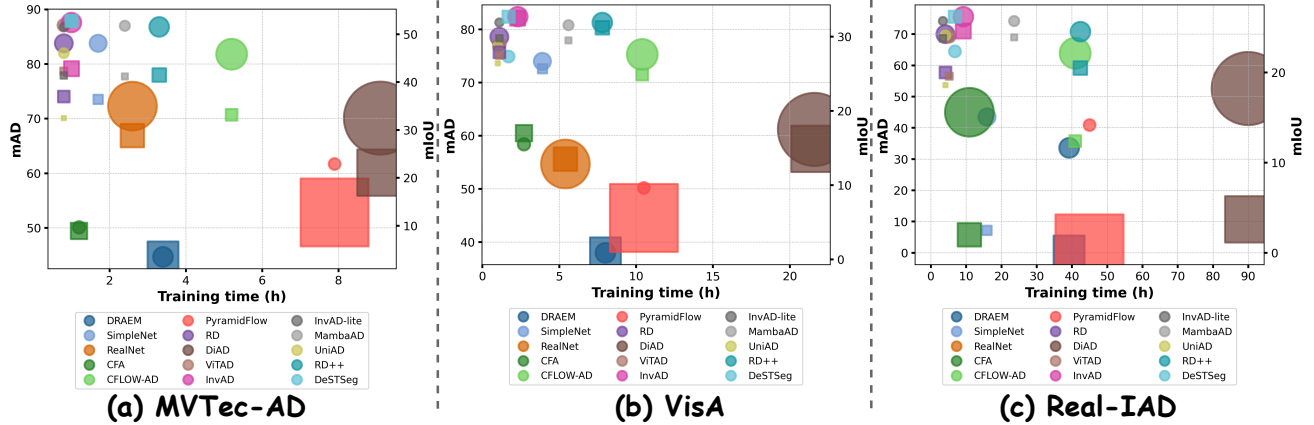


Figure 1. Intuitive benchmarked results comparison on MVTec AD [3] (Left), VisA [47] (Middle), and Real-IAD [32] (Right) datasets among mainstream methods. For each dataset, the horizontal axis represents the training time for different methods, the left vertical axis represents mAD [40] (marked as circles, with radius indicating model parameter count), and the right vertical axis represents mIoU-max [42] (marked as squares, with side length indicating model FLOPs).

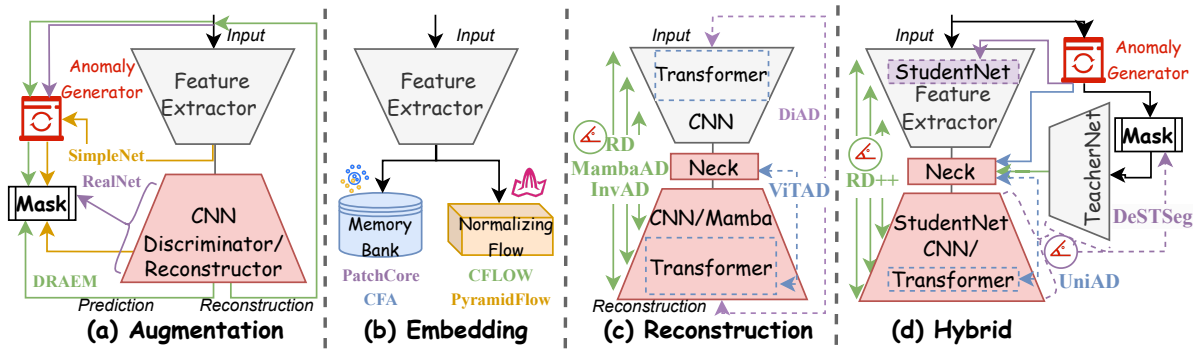


Figure 2. A comparative diagram of different frameworks for the benchmarked methods in Sec. 3.1.

utilize models pretrained on ImageNet-1K [11] as feature extractors, such as the ResNet [17] series, Wide ResNet-50 [36], and EfficientNet-b4 [30]. Recently, benefiting from the dynamic modeling capabilities of Vision Transformers (ViT) [12], some studies [9, 35, 40] have attempted to incorporate this architecture into anomaly detection.

### 3. Methodology: ADer Benchmark

#### 3.1. Supported VAD Methods

Following the categories of current VAD methods in Sec. 2.3, we choose representative models for each category. The selection criteria are based on the method’s popularity, effectiveness, and ease of use. 1) For Augmentation-based methods, we choose DRAEM [37], SimpleNet [27], and RealNet [45]. 2) For Embedding-based methods, we select CFA [21], PatchCore [29], CFLOW-AD [14], and PyramidalFlow [22]. 3) For Reconstruction-based methods, we include RD [10], DiAD [16], ViTAD [40], InvAD [42], InvAD-lite [42], and MambaAD [15]. Additionally, UniAD [35], RD++ [31], and DeSTSeg [44] are categorized as hybrid methods due to their use of multiple techniques. Fig. 2 presents schematic diagrams and com-

parisons of the frameworks for each method belonging to different types, facilitating a better understanding of the differences among these methods. Tab. 1 provides a direct comparison of the hyperparameters, efficiency, and training time on three mainstream datasets for different methods, using one L40S GPU. Note that different methods may yield varying results when tested on different hardware, but the overall relative trends remain largely unchanged.

#### 3.2. VAD Datasets

To comprehensively evaluate the effectiveness, stability, and generalization of different methods, we benchmark extensive and fair experiments on three types of datasets: 1) Real and synthetic industrial anomaly detection (AD) datasets, *i.e.*, MVTec AD [3], MVTec AD 3D [6], MVTec LOCO-AD [5], VisA [47], BTAD [28], MPDD [20], MAD.Real [46], MAD.Sim [46], and Real-IAD [32]. 2) The medical Uni-Medical [40] dataset. 3) The general-purpose COCO-AD [42] dataset. Detailed descriptions of the datasets are provided in Tab. 2, including the categories and scales of the datasets. Note that COCO-AD is inherently a multi-class dataset with four splits, and the average is taken when evaluating the comprehensive results.

	Method	Hyper Params.			Efficiency			Train Mem. (M)	Test Mem. (M)	MVTec AD		VisA		Real-IAD	
		BS	Optim.	LR	Params.	FLOPs	Backbone			Train T.	Test T.	Train T.	Test T.	Train T.	Test T.
Aug.	DRAEM [37]	8	Adam	1e-4	97.4M	198G	UNet	12,602	2,858	3.4h	35s	8.0h	36s	39.2	18m41s
	SimpleNet [27]	32	AdamW	1e-4	72.8M	17.715G	WRN50	2,266	4,946	1.7h	5m50s	3.9h	7m21s	15.9h	4h51m
	RealNet [45]	16	Adam	1e-4	591M	115G	WRN50	14,004	3,794	2.6h	41s	5.4h	41s	-	-
Emb.	CFA [21]	4	AdamW	1e-3	38.6M	55.3G	WRN50	4,364	2,826	<u>1.2h</u>	<u>18s</u>	2.7h	<u>17s</u>	10.9h	14m20s
	PatchCore [29]	8	-	-	-	-	WRN50	-	-	0.6h	9h22m	-	OOM	-	OOM
	CFLOW-AD [14]	32	Adam	2e-4	237M	28.7G	WRN50	3,048	1,892	5.2h	56s	10.4h	1m15s	40.9h	22m49s
	PyramidalFlow [22]	2	Adam	2e-4	34.3M	962G	RN18	3,904	2,836	7.9h	1m30s	10.5h	2m43s	45h	38m15s
Rec.	RD [10]	16	Adam	5e-3	80.6M	28.4G	WRN50	3,286	1,464	<b>0.8h</b>	<b>13s</b>	<u>1.1h</u>	18s	<u>4.1h</u>	<u>7m48s</u>
	DiAD [16]	12	Adam	1e-5	1331M	451.5G	RN50	26,146	20,306	9.1h	16m	21.6h	19m	90h	16h20m
	ViTAD [40]	8	AdamW	1e-4	39.0M	9.7G	ViT-S	<b>1,470</b>	<b>800</b>	<b>0.8h</b>	<u>15s</u>	<u>1.1h</u>	<b>15s</b>	<u>5.2h</u>	10m2s
	InvAD [42]	32	Adam	1e-3	95.6M	45.4G	WRN50	5,920	3,398	<u>1.0h</u>	31s	2.3h	33s	9.2h	21m
	InvAD-lite [42]	32	Adam	1e-3	<b>17.1M</b>	<u>9.3G</u>	RN34	<u>1,846</u>	<u>1,100</u>	<b>0.8h</b>	20s	<u>1.1h</u>	31s	<b>3.4h</b>	9m27s
	MambaAD [15]	16	AdamW	5e-3	<u>25.7M</u>	<u>8.3G</u>	RN34	6,542	1,484	2.4h	34s	5.6h	23s	23.6h	24m6s
Hybrid	UniAD [35]	8	AdamW	1e-4	<u>24.5M</u>	<b>3.4G</b>	EN-b4	<u>1,856</u>	<u>968</u>	<b>0.8h</b>	22s	<b>1.0h</b>	18s	<u>4.1h</u>	<b>7m2s</b>
	RD++ [31]	16	Adam	1e-3	96.1M	37.5G	WRN50	4,772	1,480	3.3h	28s	7.8h	33s	42.4h	15m17s
	DesTSeg [44]	32	SGD	0.4	35.2M	30.7G	RN18	3,446	1,240	<u>1.0h</u>	19s	<u>1.7h</u>	<u>16s</u>	6.8h	<u>8m13s</u>

Table 1. Attribute comparison for mainstream representative methods. Notations: Augmentation-based (Aug.), Embedding-based (Emb.), Reconstruction-based (Rec.), Parameters (Params), Memory (Mem.), Batch Size (BS), Optimizer (Optim.), Time (T.), ResNet (RN), Wide-ResNet (WRN), EfficientNet (EN), hours (h), minutes (m), seconds (s), unavailable (-), out-of-memory (OOM). Train and test time are evaluated under the standard setting described in Sec. 4.1 in one L40S GPU. Memory is tested under the standard setting with a batch size of 8, and the results for different methods are presented in Sec. 4.2. **Bold**, underline, and wavy-line represent the best, second-best, and third-best results, respectively.

Dataset	Category Number		Image Quantity			Epoch Setting in ADer	
	Train	Test	Train	Test		100	300
			Normal	Anomaly	Normal		
MVTec AD [3]	15	15	3,629	1,258	467	100	300
MVTec AD 3D [6]	10	10	2,950	249	948	100	300
MVTec LOCO-AD [5]	5	5	1,772	993	575	100	300
VisA [47]	12	12	8,659	962	1,200	100	300
BTAD [28]	3	3	1,799	580	451	100	300
MPDD [20]	6	6	888	282	176	100	300
MAD_Real [46]	10	10	490	221	50	100	300
MAD_Sim [46]	20	20	4,200	4,951	638	100	300
Real-IAD [32]	30	30	36,465	51,329	63,256	100	-
Uni-Medical [40]	3	3	13,339	4,499	2,514	100	300
COCO-AD [42]	61	81	30,438	1,291	3,661	100	-
			65,133	2,785	2,167	100	-
			79,083	3,328	1,624	100	-
			77,580	3,253	1,699	100	-

Table 2. Comparison of representative VAD datasets, *i.e.*, industrial, medical, and general-purpose fields, respectively. Large-scale Real-IAD and COCO-AD only employ 100 epoch setting.

### 3.3. Evaluation Metrics

Following the ViTAD [40] setting, we select image-level mean Area Under the Receiver Operating Curve (mAU-ROC) [37], mean Average Precision (mAP) [37], mean  $F_1$ -score ( $mF_1$ -max) [47], region-level mean Area Under the Per-Region-Overlap (mAU-PRO) [4], pixel-level mAU-ROC, mAP,  $mF_1$ -max, and the average AD (mAD) [40] of seven metrics to evaluate all experiments. Additionally, we adopt the more practical order-independent pixel-level mean maximal Intersection over Union (mIoU-max) proposed in InvAD [42].

### 3.4. Simplify Implementation by Structured ADer

To ensure fair comparison among different methods, we construct a standardized ADer framework. As shown in Fig. 3, it includes shared foundational training/testing components and implements various metric calculations (com-

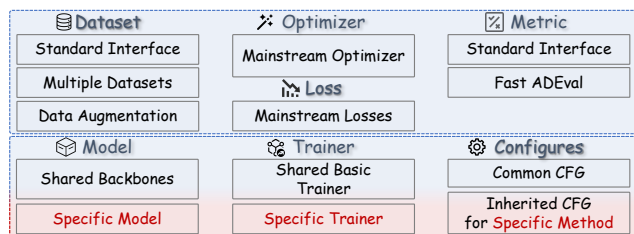


Figure 3. Core sub-modules of the framework in ADer. The blue area represents standard components, while the red area indicates that a new method requires only three corresponding files.

patible with our ADEval). The standardized dataset allows for easy comparison, eliminating potential unfair settings from different codebases. Additionally, ADer is highly extensible for new methods, requiring only compliant model, trainer, and configuration files.

### 3.5. Feature comparison with current benchmarks.

The existing vision anomaly detection benchmark works are primarily **open-iad** and **anomalib**. However, their updates for general AD models only extend up to 2022, and they **have not yet implemented the latest and practical multi-class anomaly detection methods**. We briefly discuss the relationship between the most popular Anomalib and ADer as follows: 1) From the framework perspective: Anomalib is based on PyTorch Lightning that is deeply encapsulated, whereas ADer has a shallower encapsulation, exposing more interfaces to facilitate rapid algorithm iteration. 2) From the methods perspective: Anomalib supports general AD models only up to 2022, while ADer supports more recent models up to 2024. 3) From the data and metrics perspective: Compared to Anomalib, ADer supports large-

scale industrial Real-IAD [32], medical Uni-Medical [40], and general-purpose COCO-AD [42] datasets, as well as more application-relevant metrics like mIoU-max [42] and averaged mAD [40]. 4) From the setting perspective: ADer focuses more on the recently popular and future research trend of multi-class settings.

### 3.6. ADEval: Fast and Memory-efficient Routines for mAU-ROC/mAP/mAU-PRO

The speed of metric evaluation is crucial for the iterative process of model algorithms. When the number of test images increases and the resolution becomes higher, the pixel-level evaluation algorithms implemented naively using *sklearn* and *skimage* packages become time-consuming. This is particularly evident with large-scale datasets such as Real-IAD [32] and COCO-AD [42], where evaluation times can exceed one hour. To address this issue, we have released the GPU-assisted *ADEval* library, which employs an iterative-accumulating algorithm with CUDA acceleration. By designing specialized histogram bins and employing a weighted accumulation process, the computation of metrics such as AUROC, AUPR, and AUPRO, which require sliding confidence thresholds to derive curves and then calculate the area under these curves, is optimized into an iterative-accumulative form. This approach avoids the need to cache the complete set of ground truth and predicted values during the evaluation of large-scale, high-resolution test sets.

Given a test set with inputs  $X = \{x_i\}$ , labels  $Y = \{y_i \mid i \in I\}$ , and inference results  $P_{X,Y} = \{\hat{y}_i \mid i \in I\}$ , the ROC curve  $L = \{(tpr_i, fpr_i, \hat{y}_i) \mid i \in I\}$  can be sampled as  $\hat{L} = \{(tpr_k, fpr_k, thr_k) \mid 0 \leq k < N\}$ , where  $thr_k$  is strictly increasing. The area under the ROC curve (AUROC) is given by  $\text{Trapz}(\{(fpr_i, tpr_i) \mid i \in I\}) \approx \text{Trapz}(\{fpr_k, tpr_k \mid 0 \leq k < N\})$ , where  $N$  is the number of sampling points on the ROC curve. Let  $thr_N = \max(P_{X,Y})$ .

The true positive rate (TPR) and false positive rate (FPR) are defined as follows:

$$\begin{aligned} \widehat{tpr}_k &= \frac{|\{i \mid i \in I \wedge y_i = 1 \wedge \hat{y}_i \geq thr_k\}|}{|\{i \mid i \in I \wedge y_i = 1\}|} \\ &= \frac{\sum_{l=k}^{N-1} |\{i \mid i \in I \wedge y_i = 1 \wedge thr_l \leq \hat{y}_i \leq thr_{l+1}\}|}{|\{i \mid i \in I \wedge y_i = 1\}|}, \end{aligned} \quad (1)$$

$$\begin{aligned} \widehat{fpr}_k &= \frac{|\{i \mid i \in I \wedge y_i = 0 \wedge \hat{y}_i \geq thr_k\}|}{|\{i \mid i \in I \wedge y_i = 0\}|} \\ &= \frac{\sum_{l=k}^{N-1} |\{i \mid i \in I \wedge y_i = 0 \wedge thr_l \leq \hat{y}_i \leq thr_{l+1}\}|}{|\{i \mid i \in I \wedge y_i = 0\}|}, \end{aligned} \quad (2)$$

Let  $p_k = |\{i \mid i \in I \wedge y_i = 1 \wedge thr_k \leq \hat{y}_i \leq thr_{k+1}\}|$  and  $q_k = |\{i \mid i \in I \wedge y_i = 0 \wedge thr_k \leq \hat{y}_i \leq thr_{k+1}\}|$ .

Then, the true positive rate and false positive rate can be expressed as:

$$\widehat{tpr}_k = \frac{\sum_{l=k}^{N-1} p_l}{|\{i \mid i \in I \wedge y_i = 1\}|}, \quad \widehat{fpr}_k = \frac{\sum_{l=k}^{N-1} q_l}{|\{i \mid i \in I \wedge y_i = 0\}|}, \quad (3)$$

Thus, the sampled ROC curve  $\hat{L}$  can be fully derived from  $P = \{p_k \mid 0 \leq k < N\}$  and  $Q = \{q_k \mid 0 \leq k < N\}$ . These sets  $P$  and  $Q$  can be computed iteratively and cumulatively as follows:

For  $P$ , maintain a histogram  $H = \{h_k \mid 0 \leq k < N\}$  initialized to  $h_k = 0$  for all  $0 \leq k < N$ . For each batch  $b$  of inference results  $\{\hat{y}_i \mid i \in b\}$ , update the histogram as:

$$h_k \leftarrow h_k + |\{i \mid i \in b \wedge y_i = 1 \wedge thr_k \leq \hat{y}_i \leq thr_{k+1}\}|. \quad (4)$$

Continue this process until all samples have been processed.

This method requires a space complexity of  $O(N)$ , which is significantly less than the traditional method that requires  $O(|X|)$  space to store all inference results. Additionally, each iterative update can be efficiently implemented using mature histogram operators, leveraging parallelism and GPU acceleration. This approach is particularly advantageous for computing large-scale, high-resolution evaluation metrics, especially pixel-level metrics.

## 4. Results and Analysis

### 4.1. Experimental Setup

Different methods potentially introduce various factors that can impact model performance. To ensure a fair and comprehensive evaluation of the effectiveness and convergence of different methods, we fix the most influential parameters, *i.e.*, resolution ( $256^2$ ) and training epochs (100 and 300). The reason lies in the fact that tasks such as classification, detection, and segmentation typically set specific resolutions and standard training epochs. We observe that for most methods, 100 epochs generally suffice to reach saturation [40, 42], with only a few methods [35] requiring more epochs for training. Therefore, we also establish a setting with 300 epochs. Meanwhile, we maintain consistency with the original papers for batch size, optimizer, learning rate, and data augmentation. We report the evaluation results corresponding to the final epoch at the end of training to ensure fairness. All experiments are conducted on one L40S GPU.

### 4.2. Benchmark Results on Industrial, Medical, and General-purpose UAD Datasets

To thoroughly evaluate the effectiveness of different methods and their adaptability to various data domains, we conduct experiments on multiple datasets across three domains.

Table 3. Benchmarked results on MVTec AD dataset [3] by the suggested metrics in Sec. 3.3 under 100/300 epochs. **Bold**, underline, and wavy-line represent the best, second-best, and third-best results, respectively. Patchcore requires no training that shares results under different epoch settings.

	Method	Image-level			Pixel-level			mAU-PRO	mIoU-max	mAD
		mAU-ROC	mAP	$mF_1$ -max	mAU-ROC	mAP	$mF_1$ -max			
Aug.	DRAEM [37]	54.5/55.2	76.3/77.0	83.6/83.9	47.6/48.7	3.2/ 3.1	6.7/ 6.3	14.3/15.8	3.5/ 3.3	40.9/41.4
	SimpleNet [27]	95.4/79.2	98.3/90.8	95.7/87.6	96.8/82.4	48.8/24.0	51.9/29.0	86.9/62.0	36.4/17.8	82.0/65.0
	RealNet [45]	84.8/82.9	94.1/93.3	90.9/90.9	72.6/69.8	48.2/50.0	41.4/40.4	56.8/51.2	28.8/28.5	69.8/68.4
Emb.	CFA [21]	57.6/55.8	78.3/78.8	84.7/84.5	54.8/43.9	11.9/ 4.8	14.7/ 8.9	25.3/19.3	8.9/ 4.7	46.8/42.3
	PatchCore [29]	<b>98.8/ -</b>	<b>99.5/ -</b>	<b>98.4/ -</b>	<b>98.3/ -</b>	<u>59.9/ -</u>	<u>61.0/ -</u>	<u>94.2/ -</u>	<u>44.9/ -</u>	<b>87.2/ -</b>
	CFLOW-AD [14]	91.6/92.7	96.7/97.2	93.4/94.0	95.7/95.8	45.9/46.8	48.6/49.6	88.3/89.0	33.2/34.0	80.0/80.7
	PyramidalFlow [22]	70.2/66.2	85.5/84.3	85.5/85.1	80.0/74.2	22.3/17.2	22.0/19.6	47.5/40.0	12.8/11.4	59.0/55.2
Rec.	RD [10]	93.6/90.5	97.2/95.0	95.6/95.1	95.8/95.9	48.2/47.1	53.6/52.1	91.2/91.2	37.0/35.8	82.2/81.0
	DiAD [16]	88.9/92.0	95.8/96.8	93.5/94.4	89.3/89.3	27.0/27.3	32.5/32.7	63.9/64.4	21.1/21.3	70.1/71.0
	ViTAD [40]	<u>98.3/98.4</u>	<u>99.3/99.4</u>	<u>97.3/97.5</u>	<u>97.6/97.5</u>	<u>55.2/55.2</u>	<u>58.4/58.1</u>	<u>92.0/91.7</u>	<u>42.3/42.0</u>	<u>85.4/85.4</u>
	InvAD [42]	<u>98.1/98.9</u>	<u>99.0/99.6</u>	<u>97.6/98.2</u>	<u>98.0/98.1</u>	<u>56.3/57.1</u>	<u>59.2/59.6</u>	<b>94.4/94.4</b>	<u>42.8/43.1</u>	<u>86.1/86.6</u>
	InvAD-lite [42]	97.9/98.1	99.2/99.1	96.8/96.8	97.3/97.3	54.4/55.0	57.8/58.1	93.3/93.1	41.4/41.7	85.2/85.4
	MambaAD [15]	97.8/98.5	<u>99.3/99.5</u>	<u>97.3/97.7</u>	97.4/97.6	<u>55.1/56.1</u>	<u>57.6/58.7</u>	<u>93.4/93.6</u>	<u>41.2/42.3</u>	<u>85.4/86.0</u>
		UniAD [35]	92.5/96.8	97.3/98.9	95.4/97.0	95.8/96.8	42.7/45.0	48.0/50.2	89.3/91.0	32.5/34.2
Hybrid	RD++ [31]	97.9/95.8	98.8/98.0	96.4/96.6	97.3/97.3	54.7/53.0	58.0/57.0	93.2/92.9	41.5/40.5	85.2/84.4
	DesTSeg [44]	96.4/96.3	98.6/98.8	96.2/96.1	92.0/92.6	<b>71.1/75.8</b>	<b>68.2/71.3</b>	83.4/82.6	<b>52.8/56.6</b>	<u>86.6/87.6</u>

Table 4. Benchmarked results on VisA dataset [47] by the suggested metrics under 100/300 epochs.

	Method	Image-level			Pixel-level			mAU-PRO	mIoU-max	mAD
		mAU-ROC	mAP	$mF_1$ -max	mAU-ROC	mAP	$mF_1$ -max			
Aug.	DRAEM [37]	55.1/56.2	62.4/64.6	72.9/74.9	37.5/45.0	0.6/ 0.7	1.7/ 1.8	10.0/16.0	0.9/ 0.9	34.3/37.0
	SimpleNet [27]	86.4/80.7	89.1/83.8	82.8/79.3	96.6/94.4	34.0/29.2	37.8/33.1	79.2/74.2	25.7/22.1	72.3/67.8
	RealNet [45]	71.4/79.2	79.5/84.8	74.7/78.3	61.0/65.4	25.7/29.2	22.6/27.9	27.4/33.9	13.5/17.4	51.8/57.0
Emb.	CFA [21]	66.3/67.1	74.3/73.8	74.2/75.3	81.3/83.0	22.1/13.7	26.2/18.7	50.8/48.7	17.0/11.3	56.5/54.3
	CFLOW-AD [14]	86.5/87.2	88.8/89.5	84.9/85.1	97.7/97.8	33.9/34.2	37.2/37.2	86.8/87.3	24.9/24.9	73.7/74.0
	PyramidalFlow [22]	58.2/69.0	66.3/72.9	74.4/75.8	77.0/79.1	7.2/ 7.9	9.6/ 8.7	42.8/52.6	5.6/ 4.7	47.9/52.3
Rec.	RD [10]	90.6/93.9	90.9/94.8	89.3/90.4	98.0/98.1	35.4/38.4	42.5/43.7	<u>91.9/91.9</u>	27.9/29.0	76.9/78.7
	DiAD [16]	84.8/90.5	88.5/91.4	86.9/90.4	82.5/83.4	17.9/19.2	23.2/25.0	44.5/44.3	14.9/16.2	61.2/63.5
	ViTAD [40]	90.4/90.3	91.1/91.2	86.0/86.4	98.2/98.2	36.4/36.4	41.0/40.9	85.7/85.8	27.5/27.5	75.5/75.6
	InvAD [42]	<b>95.4/95.6</b>	<b>95.7/96.0</b>	<b>91.6/92.3</b>	<b>98.9/99.0</b>	<u>43.3/43.7</u>	<u>46.8/46.9</u>	<u>93.1/93.0</u>	<u>32.5/32.6</u>	<b>80.7/80.9</b>
	InvAD-lite [42]	<u>94.9/95.3</u>	<u>95.2/95.8</u>	<u>90.7/91.0</u>	<u>98.6/98.7</u>	<u>40.2/41.2</u>	<u>44.0/44.9</u>	<b>93.1/93.2</b>	<u>29.8/30.6</u>	<u>79.5/80.0</u>
	MambaAD [15]	<u>94.5/93.6</u>	<u>94.9/93.9</u>	<u>90.2/89.8</u>	<u>98.4/98.2</u>	39.3/34.0	43.7/39.3	<u>92.1/90.5</u>	29.5/25.9	79.0/77.0
Hybrid	UniAD [35]	89.0/91.4	91.0/93.3	85.8/87.5	98.3/98.5	34.5/35.3	39.6/40.2	86.5/89.0	26.4/26.5	75.0/76.5
	RD++ [31]	93.9/93.1	94.7/94.1	<u>90.2/90.0</u>	98.4/98.4	<u>42.3/40.4</u>	<u>46.3/44.8</u>	<u>91.9/91.4</u>	<u>31.2/29.9</u>	<u>79.7/78.9</u>
	DesTSeg [44]	89.9/89.0	91.4/90.3	86.7/85.9	86.7/84.8	<u>46.6/43.3</u>	<u>47.2/44.4</u>	61.1/57.5	<u>32.7/30.1</u>	72.8/70.7

Due to space constraints, we report the average metrics for the popular MVTec AD (see Tab. 3), VisA (see Tab. 4), and Real-IAD (see Tab. 5) datasets in the main paper. For the remaining datasets, we report the mAD and mIoU-max metrics (see Tab. 6). Full results for each category are provided in Appendix A to V.

**Quantitative results.** InvAD [42] consistently shows excellent performance across all datasets. ViTAD [40] and MambaAD [15], specifically designed for multi-class settings, also achieve good results. In contrast, DiAD [16] and UniAD [35] require more epochs to converge and do not perform well under the 100/300 epoch standard we set. DeTSeg [44] exhibits outstanding performance in pixel-level segmentation. Methods designed for single-class settings, such as RD [10], RD++ [31], CFLOW-AD [14], and RealNet [45], also perform well in multi-class settings. However, single-class methods like DRAEM [37], SimpleNet [27], CFA [21], and PyramidFlow [22] show sig-

nificant performance gaps in multi-class anomaly detection and are not suitable for such tasks. Considering the training time, model parameters, and FLOPs shown in Fig. 2, InvAD, InvAD-lite, and ViTAD achieve a good balance of effectiveness and efficiency. RD, UniAD, and DeTSeg also perform well in terms of both efficiency and effectiveness. On the other hand, methods like DiAD, PyramidFlow, CFLOW-AD, RD++, RealNet, MambaAD, and SimpleNet have significantly longer training times compared to others. **Qualitative results.** Fig. 4 in Appendix presents intuitive visualization results under the 100 epochs training setting on popular MVTec AD [3] and VisA [47] datasets, as well as the medical Uni-Medical [40] and large-scale Real-IAD [32] datasets.

**Convergence analysis.** From Tab. 3 and Tab. 4, we analyze the convergence of different methods by comparing the results after training for 100/300 epochs. The methods can be categorized into three groups: 1) Methods that show no sig-

Table 5. Benchmarked results on Real-IAD dataset [32] by the suggested metrics under 100 epoch.

	Method	Image-level			Pixel-level			mAU-PRO	mIoU-max	mAD
		mAU-ROC	mAP	$mF_1$ -max	mAU-ROC	mAP	$mF_1$ -max			
Aug.	DRAEM [37]	50.9	45.9	61.3	44.0	0.2	0.4	13.6	0.2	30.9
	SimpleNet [27]	54.9	50.6	61.5	76.1	1.9	4.9	42.4	2.5	41.8
Emb.	CFA [21]	55.7	50.5	61.9	81.3	1.6	3.8	48.8	2.0	43.4
	CFLOW-AD [14]	77.0	75.8	69.9	94.8	17.6	21.7	80.4	12.4	62.5
	PyramidalFlow [22]	54.4	48.0	62.0	71.1	1.2	1.1	34.9	0.5	39.0
Rec.	RD [10]	82.7	79.3	74.1	97.2	25.2	32.8	90.0	20.0	68.8
	DiAD [16]	75.6	66.4	69.9	88.0	2.9	7.1	58.1	3.7	52.6
	ViTAD [40]	82.7	80.2	73.7	97.2	24.3	32.3	84.8	19.6	67.9
	InvAD [42]	<b>89.4</b>	<b>87.0</b>	<b>80.2</b>	<u>98.4</u>	<u>32.6</u>	<u>38.9</u>	<b>92.7</b>	<u>24.6</u>	<b>74.2</b>
	InvAD-lite [42]	<u>87.2</u>	<u>85.2</u>	<u>77.8</u>	<u>98.0</u>	<u>31.7</u>	<u>37.9</u>	<u>92.0</u>	<u>23.8</u>	<u>72.8</u>
	MambaAD [15]	<u>87.0</u>	<u>85.3</u>	<u>77.6</u>	<b>98.6</b>	<u>32.4</u>	<u>38.1</u>	<u>91.2</u>	<u>23.9</u>	<u>72.9</u>
Hybrid	UniAD [35]	83.1	81.2	74.5	97.4	23.3	30.9	87.1	18.6	68.2
	RD++ [31]	83.6	80.6	74.8	97.7	25.9	33.6	90.7	20.5	69.6
	DesTSeg [44]	79.3	76.7	70.7	80.3	<b>36.9</b>	<b>40.3</b>	56.1	<b>26.2</b>	62.9

Table 6. Benchmarked results on all other datasets by the mIoU/mAD metrics under 100 epoch.

	Method	MVTec 3D	MVTec LOCO	BTAD	MPDD	MAD_Real	MAD_Sim	Uni-Medical	COCO-AD
Aug.	DRAEM [37]	1.0/42.5	5.6/38.5	3.4/43.3	2.5/32.8	0.8/43.8	0.7/48.0	3.0/33.5	8.0/36.0
	SimpleNet [27]	13.9/67.8	<u>21.2/65.0</u>	28.6/74.3	24.5/76.2	6.3/54.3	4.2/58.0	23.3/67.7	11.5/41.0
	RealNet [45]	- / -	- / -	36.6/73.7	28.2/70.1	- / -	- / -	- / -	- / -
Emb.	CFA [21]	9.3/55.2	9.1/49.0	33.6/78.2	16.6/62.3	8.7/56.5	4.6/55.1	14.7/55.0	8.9/38.7
	PatchCore [29]	24.5/76.7	20.4/65.4	38.0/81.5	<b>35.0/81.9</b>	<u>16.6/66.6</u>	- / -	- / -	- / -
	CFLOW-AD [14]	15.8/69.8	17.3/61.7	33.8/77.7	20.1/68.3	<u>8.8/61.2</u>	2.7/57.1	17.7/68.6	<u>16.0/51.5</u>
	PyramidalFlow [22]	6.4/59.7	8.0/44.7	18.3/66.1	10.4/62.9	5.1/56.7	2.5/54.5	9.4/45.7	8.0/36.2
Rec.	RD [10]	22.2/73.7	15.8/60.6	42.1/83.2	31.4/79.0	7.2/58.2	4.5/60.0	26.9/70.3	11.5/44.3
	DiAD [16]	5.4/62.8	14.9/56.5	15.7/68.5	8.2/58.1	3.6/55.8	4.2/58.9	23.2/69.1	11.6/44.1
	ViTAD [40]	20.4/73.5	19.8/62.5	40.1/81.5	27.7/75.6	5.0/55.2	<u>5.0/60.1</u>	<b>33.7/74.7</b>	<b>20.1/52.2</b>
	InvAD [42]	<u>27.4/78.6</u>	<b>23.1/67.0</b>	<b>44.3/84.5</b>	<u>34.0/80.2</u>	<b>16.8/64.7</b>	<b>8.6/67.4</b>	<u>32.6/74.6</u>	<u>14.3/49.0</u>
	InvAD-lite [42]	<u>26.9/78.0</u>	20.6/64.9	<u>42.6/82.9</u>	30.9/78.2	8.1/60.2	<u>6.0/62.1</u>	26.5/70.7	13.7/47.6
	MambaAD [15]	<u>25.9/77.4</u>	20.6/64.3	39.0/80.9	26.8/76.3	7.2/58.2	<u>5.0/60.8</u>	<b>33.5/75.0</b>	12.9/47.1
Hybrid	UniAD [35]	16.7/70.3	<u>21.6/62.4</u>	36.9/81.1	12.5/61.7	5.8/56.1	3.5/58.3	27.6/70.7	10.9/42.4
	RD++ [31]	25.2/76.4	17.6/62.2	<u>42.8/83.3</u>	<u>33.6/79.3</u>	8.5/59.2	4.4/60.0	29.4/71.6	11.8/45.0
	DesTSeg [44]	<b>28.4/70.4</b>	20.3/61.3	29.0/74.6	25.6/69.3	4.5/50.1	4.1/49.7	21.2/58.6	8.5/38.6

nificant improvement in performance after 300 epochs compared to 100 epochs, indicating rapid convergence within 100 epochs. These models include DRAEM [37], SimpleNet [27], CFA [21], PyramidFlow [22], ViTAD [40], InvAD-lite [42], RD++ [31], and DeSTSeg [44]. 2) Methods that show improvement with continued training on the VisA dataset but no improvement or a decline on the MVTec AD dataset, indicating slower convergence on larger datasets. These models include RealNet and RD. 3) Methods that show significant improvement after 300 epochs compared to 100 epochs, indicating slower convergence. These models include CFLOW-AD [14], DiAD [16], InvAD [42], MambaAD [15], and UniAD [35].

**Stability analysis.** For current anomaly detection algorithms, most authors select the best epoch’s results as the model’s performance. However, this method of epoch selection is unscientific and may indicate significant model instability. Therefore, we further analyze model stability using Tab. 3 and Tab. 4, comparing the results at 100/300 epochs to identify any substantial differences. The results show that SimpleNet [27] and PyramidFlow [22] exhibit considerable differences, indicating poor model stability,

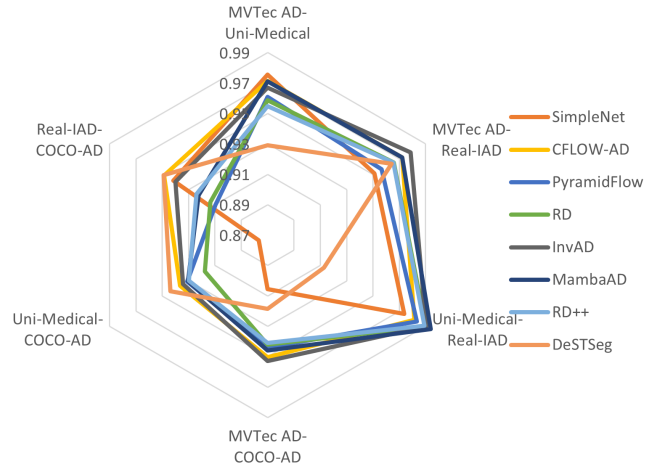


Figure 4. A Pearson correlation coefficient analysis for different methods on several datasets.

while other methods do not show significant fluctuations. **Cross-domain dataset correlation.** To analyze the adaptability of different methods across various datasets and the relationships and differences between different types of datasets, we employ Pearson correlation analysis to exam-

ine the correlations among these datasets. Specifically, we select four distinct datasets for analysis: MVTec AD [3], Uni-Medical [40], Real-IAD [32], and COCO-AD [42]. MVTec AD represents a fundamental industrial dataset, Uni-Medical consists of medical images from CT scans, Real-IAD is a large-scale multi-view industrial dataset from real-world scenarios, and COCO-AD is a large-scale panoptic segmentation dataset from real-life scenes. We evaluate four categories of methods using eight metrics: image and pixel mAU-ROC, mAP, mF1-max, region mAU-PRO, and segmentation mIoU-max. The results, as shown in Fig. 4, indicate that the COCO-AD and Uni-Medical datasets exhibit lower Pearson correlation coefficients due to significant differences in data distribution compared to general industrial datasets. Although the Pearson correlation coefficient between the Uni-Medical and Real-IAD datasets is relatively high, Tab. 5 and Tab. A11 in the Appendix reveal that this is because all methods perform poorly on these two datasets. Additionally, it is observed that the methods SimpleNet and DeSTSeg show considerable instability in their results across different datasets. This instability may be attributed to the inherent instability of the data augmentation algorithms they employ.

**Training-free PatchCore.** PatchCore [29] does not require model training. It extracts all features from the training data, then selects a core subset and stores it in a Memory Bank. During testing, each test image is compared with the Memory Bank to compute an anomaly score. Because it stores the core subset of all normal features in the Memory Bank, PatchCore is only feasible for multi-class anomaly detection tasks on small-scale datasets. For large-scale datasets, it faces limitations due to insufficient GPU and memory resources. Although it achieves excellent results on the MVTec AD dataset, as shown in Tab. 1, its testing time is nearly a thousand times longer than other methods. In summary, PatchCore performs exceptionally well on small-scale datasets but is constrained by large-scale datasets and testing time.

**Dataset Analysis.** The experimental results indicate that there is room for improvement in the VisA [47] and Real-IAD [32] datasets due to the very small defect areas, necessitating models with stronger capabilities for detecting minor defects. The MAD\_Real and MAD\_Sim [46] datasets, due to their small data volume and varying difficulty levels, result in similar performance across all models, particularly in the mF1-max metric. The Uni-Medical [40] dataset, consisting of images converted from CT scans, has a data distribution that significantly differs from other industrial datasets, suggesting the need for specialized detection networks tailored to medical datasets. COCO-AD [42], as a newly proposed large-scale dataset for general scenarios, presents high complexity. Current industrial AD networks are unable to achieve effective results on the COCO-AD.

### 4.3. Challenges for Current VAD

**Immature method.** For challenging anomaly detection datasets such as MVTecLOCO, pose-agnostic MAD, and general-purpose COCO-AD, current methods perform poorly in a multi-class setting. Future research should focus on designing more robust methods to address this issue.

**Efficiency.** Most methods do not consider model complexity during design, resulting in high FLOPs. This issue becomes more pronounced when applied to real-world high-resolution scenarios. Incorporating lightweight characteristics in model design could be a potential solution.

**Dataset scale.** Mainstream datasets in the VAD field, such as MVTec AD and Real-IAD, are relatively small compared to those in detection and segmentation fields and are tailored to specific industrial scenarios. This limitation could hinder technological development. Collecting larger-scale, general-scene AD datasets is crucial for the advancement of the VAD field.

**VAD-specific metric.** Metrics like mAU-ROC and mAP are not uniquely designed for the BAD field. Developing more reliable evaluation methods to better meet practical application needs is essential.

**Augmentation and tricks.** In fields such as classification, detection, and segmentation, data augmentation and tricks are extremely important for model training. However, few studies explore their role in the AD field, potentially limiting model performance.

**Model interpretability.** In many applications, understanding why a model detects a particular anomaly is crucial.

## 5. Conclusion and Discussion

This paper addresses the urgent need for a comprehensive and fair benchmark in the field of visual anomaly detection. We introduce a modular and scalable *ADer* library designed to fairly facilitate the evaluation of fifteen advanced VAD methods across multiple mainstream datasets, ensuring a thorough and unbiased assessment of each method’s performance. Our extensive experiments reveal the strengths and weaknesses of different methods, providing valuable insights into their efficiency and training resource consumption. We also develop and open-source a GPU-assisted *ADEval* package to reduce the evaluation time, enabling extensive assessments. Experimental results highlight the challenges of various VAD methods and offer valuable insights for future research directions.

**Broader Impacts.** The open-sourcing *ADer* can accelerate the development of new VAD technology for the open-source community and become a valuable resource for practitioners in the field.

## References

- [1] Samet Akçay, Amir Atapour-Abarghouei, and Toby P Breckon. Ganomaly: Semi-supervised anomaly detection via adversarial training. In *ACCV*, 2019. 2
- [2] Samet Akçay, Amir Atapour-Abarghouei, and Toby P Breckon. Skip-ganomaly: Skip connected and adversarially trained encoder-decoder anomaly detection. In *IJCNN*, 2019. 2
- [3] Paul Bergmann, Michael Fauser, David Sattlegger, and Carsten Steger. Mvtec ad—a comprehensive real-world dataset for unsupervised anomaly detection. In *CVPR*, 2019. 3, 4, 6, 8
- [4] Paul Bergmann, Michael Fauser, David Sattlegger, and Carsten Steger. Uninformed students: Student-teacher anomaly detection with discriminative latent embeddings. In *CVPR*, 2020. 2, 4
- [5] Paul Bergmann, Kilian Batzner, Michael Fauser, David Sattlegger, and Carsten Steger. Beyond dents and scratches: Logical constraints in unsupervised anomaly detection and localization. *IJCV*, 2022. 3, 4
- [6] Paul Bergmann, Xin Jin, David Sattlegger, and Carsten Steger. The mvtec 3d-ad dataset for unsupervised 3d anomaly detection and localization. In *VISIGRAPP*, 2022. 3, 4
- [7] 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
- [8] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *ICCV*, 2021. 2
- [9] Axel De Nardin, Pankaj Mishra, Gian Luca Foresti, and Claudio Piciarelli. Masked transformer for image anomaly localization. *IJNS*, 2022. 3
- [10] Hanqiu Deng and Xingyu Li. Anomaly detection via reverse distillation from one-class embedding. In *CVPR*, 2022. 2, 3, 4, 6, 7
- [11] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 3
- [12] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021. 3
- [13] Zhihao Gu, Liang Liu, Xu Chen, Ran Yi, Jiangning Zhang, Yabiao Wang, Chengjie Wang, Annan Shu, Guannan Jiang, and Lizhuang Ma. Remembering normality: Memory-guided knowledge distillation for unsupervised anomaly detection. In *ICCV*, 2023. 2
- [14] Denis Gudovskiy, Shun Ishizaka, and Kazuki Kozuka. Cflow-ad: Real-time unsupervised anomaly detection with localization via conditional normalizing flows. In *CACV*, 2022. 2, 3, 4, 6, 7
- [15] Haoyang He, Yuhu Bai, Jiangning Zhang, Qingdong He, Hongxu Chen, Zhenye Gan, Chengjie Wang, Xiangtai Li, Guanzhong Tian, and Lei Xie. Mambaad: Exploring state space models for multi-class unsupervised anomaly detection. *NeurIPS*, 2024. 3, 4, 6, 7
- [16] Haoyang He, Jiangning Zhang, Hongxu Chen, Xuhai Chen, Zhishan Li, Xu Chen, Yabiao Wang, Chengjie Wang, and Lei Xie. A diffusion-based framework for multi-class anomaly detection. In *AAAI*, 2024. 2, 3, 4, 6, 7
- [17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 2, 3
- [18] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *CVPR*, 2022. 2
- [19] Teng Hu, Jiangning Zhang, Ran Yi, Yuzhen Du, Xu Chen, Liang Liu, Yabiao Wang, and Chengjie Wang. Anomalydiffusion: Few-shot anomaly image generation with diffusion model. In *AAAI*, 2024. 2
- [20] Stepan Jezek, Martin Jonak, Radim Burget, Pavel Dvorak, and Milos Skotak. Deep learning-based defect detection of metal parts: evaluating current methods in complex conditions. In *ICUMT*, 2021. 3, 4
- [21] Sungwook Lee, Seunghyun Lee, and Byung Cheol Song. Cfa: Coupled-hypersphere-based feature adaptation for target-oriented anomaly localization. *IEEE Access*, 2022. 3, 4, 6, 7
- [22] Jiarui Lei, Xiaobo Hu, Yue Wang, and Dong Liu. Pyramid-flow: High-resolution defect contrastive localization using pyramid normalizing flow. In *CVPR*, 2023. 2, 3, 4, 6, 7
- [23] Chun-Liang Li, Kihyuk Sohn, Jinsung Yoon, and Tomas Pfister. Cutpaste: Self-supervised learning for anomaly detection and localization. In *CVPR*, 2021. 2
- [24] Yufei Liang, Jiangning Zhang, Shiwei Zhao, Runze Wu, Yong Liu, and Shuwen Pan. Omni-frequency channel-selection representations for unsupervised anomaly detection. *TIP*, 2023. 2
- [25] Jiaqi Liu, Guoyang Xie, Jinbao Wang, Shangnian Li, Chengjie Wang, Feng Zheng, and Yaochu Jin. Deep industrial image anomaly detection: A survey. *MIR*, 2024. 1
- [26] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, 2021. 2
- [27] Zhikang Liu, Yiming Zhou, Yuansheng Xu, and Zilei Wang. Simplenet: A simple network for image anomaly detection and localization. In *CVPR*, 2023. 2, 3, 4, 6, 7
- [28] Pankaj Mishra, Riccardo Verk, Daniele Fornasier, Claudio Piciarelli, and Gian Luca Foresti. Vt-adl: A vision transformer network for image anomaly detection and localization. In *ISIE*, 2021. 3, 4
- [29] Karsten Roth, Latha Pemula, Joaquin Zepeda, Bernhard Schölkopf, Thomas Brox, and Peter Gehler. Towards total recall in industrial anomaly detection. In *CVPR*, 2022. 2, 3, 4, 6, 7, 8
- [30] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *ICML*, 2019. 3
- [31] Tran Dinh Tien, Anh Tuan Nguyen, Nguyen Hoang Tran, Ta Duc Huy, Soan Duong, Chanh D Tr Nguyen, and

- Steven QH Truong. Revisiting reverse distillation for anomaly detection. In *CVPR*, 2023. 2, 3, 4, 6, 7
- [32] Chengjie Wang, Wenbing Zhu, Bin-Bin Gao, Zhenye Gan, Jianning Zhang, Zhihao Gu, Shuguang Qian, Mingang Chen, and Lizhuang Ma. Real-iad: A real-world multi-view dataset for benchmarking versatile industrial anomaly detection. In *CVPR*, 2024. 1, 3, 4, 5, 6, 7, 8
- [33] Shenzhi Wang, Liwei Wu, Lei Cui, and Yujun Shen. Glancing at the patch: Anomaly localization with global and local feature comparison. In *CVPR*, 2021. 2
- [34] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In *ICCV*, 2021. 2
- [35] Zhiyuan You, Lei Cui, Yujun Shen, Kai Yang, Xin Lu, Yu Zheng, and Xinyi Le. A unified model for multi-class anomaly detection. *NeurIPS*, 2022. 2, 3, 4, 5, 6, 7
- [36] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In *BMVC*, 2016. 3
- [37] Vitjan Zavrtanik, Matej Kristan, and Danijel Skočaj. Draem: a discriminatively trained reconstruction embedding for surface anomaly detection. In *ICCV*, 2021. 2, 3, 4, 6, 7
- [38] Gongjie Zhang, Kaiwen Cui, Tzu-Yi Hung, and Shijian Lu. Defect-gan: High-fidelity defect synthesis for automated defect inspection. In *CACV*, 2021. 2
- [39] Jiangning Zhang, Chao Xu, Jian Li, Wenzhou Chen, Yabiao Wang, Ying Tai, Shuo Chen, Chengjie Wang, Feiyue Huang, and Yong Liu. Analogous to evolutionary algorithm: Designing a unified sequence model. *NeurIPS*, 2021. 2
- [40] Jiangning Zhang, Xuhai Chen, Yabiao Wang, Chengjie Wang, Yong Liu, Xiangtai Li, Ming-Hsuan Yang, and Dacheng Tao. Exploring plain vit reconstruction for multi-class unsupervised anomaly detection. *arXiv preprint arXiv:2312.07495*, 2023. 1, 2, 3, 4, 5, 6, 7, 8
- [41] Jiangning Zhang, Xiangtai Li, Jian Li, Liang Liu, Zhucun Xue, Boshen Zhang, Zhengkai Jiang, Tianxin Huang, Yabiao Wang, and Chengjie Wang. Rethinking mobile block for efficient attention-based models. In *ICCV*, 2023. 2
- [42] Jiangning Zhang, Xiangtai Li, Guanzhong Tian, Zhucun Xue, Yong Liu, Guansong Pang, and Dacheng Tao. Learning feature inversion for multi-class unsupervised anomaly detection under general-purpose coco-ad benchmark. *arXiv*, 2024. 1, 2, 3, 4, 5, 6, 7, 8
- [43] Jiangning Zhang, Xiangtai Li, Yabiao Wang, Chengjie Wang, Yibo Yang, Yong Liu, and Dacheng Tao. Eatformer: Improving vision transformer inspired by evolutionary algorithm. *IJCV*, 2024. 2
- [44] Xuan Zhang, Shiyu Li, Xi Li, Ping Huang, Jiulong Shan, and Ting Chen. Destseg: Segmentation guided denoising student-teacher for anomaly detection. In *CVPR*, 2023. 2, 3, 4, 6, 7
- [45] Ximiao Zhang, Min Xu, and Xiuzhuang Zhou. Realnet: A feature selection network with realistic synthetic anomaly for anomaly detection. In *CVPR*, 2024. 3, 4, 6, 7
- [46] Qiang Zhou, Weize Li, Lihan Jiang, Guoliang Wang, Guyue Zhou, Shanghang Zhang, and Hao Zhao. Pad: A dataset and benchmark for pose-agnostic anomaly detection. *NeurIPS*, 2024. 3, 4, 8
- [47] Yang Zou, Jongheon Jeong, Latha Pemula, Dongqing Zhang, and Onkar Dabeer. Spot-the-difference self-supervised pre-training for anomaly detection and segmentation. In *ECCV*, 2022. 3, 4, 6, 8