AD^2: Anomaly Detection During Training an Distillation-Based Anomaly Detection Model

Kai Chen[1,2], Xiaowang Wang[1,2], Huiyue Yang[1,2], Hui Chen[1,2], Yuwang Wang[1,2], Sicheng Zhao[1,2], and Guiguang Ding[1,2(✉)]

[1] Tsinghua University, Beijing 100084, China
wang-yuwang@mail.tsinghua.edu.cn
[2] Beijing National Research Center for Information Science and Technology (BNRist), Beijing, China
dinggg@tsinghua.edu.cn

Abstract. Anomaly Detection (AD) technology has received much attention recently, especially in industrial quality inspection applications. Most existing unsupervised AD methods assume that the training data contains only normal samples, which is difficult to satisfy in practice. When the training data are mixed with even a small number of defective samples, the AD methods, that use distillation learning, will be negatively affected, leading to significant performance drops. To tackle this issue, in this paper, we proposed an approach, namely AD^2, to conduct anomaly detection during the training phase of an anomaly detection model. Specifically, we devise a Non-Major Feature Elimination (NMFE) module to eliminate the prominent anomaly-related discrepancy information and adopt an Anomaly Training Data Removal (ATDR) strategy to identify outliers in the training data, preventing abnormal information from affecting model training. During the inference phase, AD^2 does not introduce any extra computation overhead. Experiments demonstrate that AD^2 can successfully alleviate the performance deterioration caused by polluted training samples. On the MVTec LOCO dataset, when 10% of the training set is corrupted by anomalous samples, AD^2 can significantly improve the image-level AUROC from 0.793 to 0.865 compared to the ordinary AD method, without sacrificing any inference efficiency. AD^2 provides an effective solution for issues of data uncertainty in anomaly detection. The source code will be released.

1 Introduction

The unsupervised Anomaly Detection (AD)[1] task that aims to detect anomaly samples given a large number of normal data. AD methods can be utilized by

This work was supported by the Key R&D Program of Xinjiang, China (2022B01006); National Natural Science Foundation of China (No. 62021002); the China Postdoctoral Science Foundation under Grant Number 2025M773465; and the Postdoctoral Fellowship Program of CPSF (GZC20231319).

[1] In this paper, AD refers to the unsupervised image Anomaly Detection.

J. Kittler et al. (Eds.): PRCV 2025, LNCS 16283, pp. 3–17, 2026.
https://doi.org/10.1007/978-981-95-5761-5_1

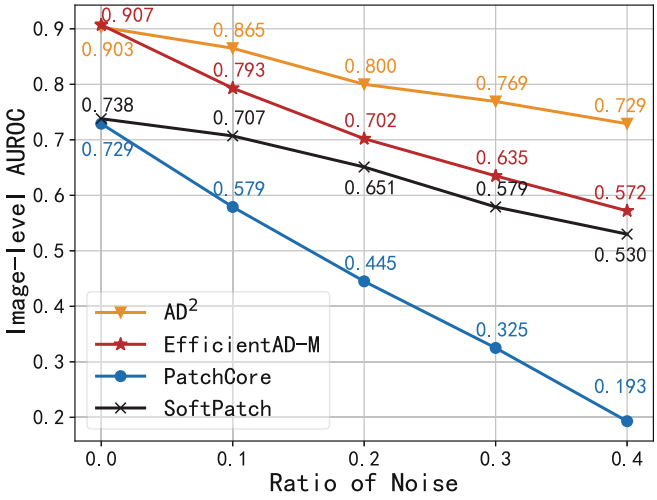

Fig. 1. Comparison of image-level AUROC performance on polluted MVTec LOCO dataset at different ratios of noise. The performance of ordinary AD methods deteriorates significantly as the ratio of noise increases. Compared to them, our proposed AD^2 is more robust and has a higher tolerance for noise. In practical scenarios where the training data may contain anomalies, AD^2 can achieve better results.

industrial manufacturers to ensure production quality [24]. They can also be used in medical diagnosis to find rare underlying disease representations [25]. Compared to supervised learning methods, AD methods require less expensive labeling labor from experts but have the ability to detect unseen defects [22]. These advantages make AD methods receive wide attention from computer vision researchers [11]. However, traditional AD methods assume that the training data only contain normal samples, which is not easy to satisfy in practical situations [12]. Completely clean data sources are rarely available, which hinders the application in practical fields. For example, the data is often collected directly from industrial production lines, which is a "blind" scenario described in [30]. Without manual confirmation, the purity of normal data cannot be guaranteed.

Most AD methods do not consider the negative impact of noisy training data. Abnormal samples mixed into the training data may mislead the AD methods to capture incorrect information during the training phase, resulting in significant performance deterioration at the testing stage [12]. As shown in Fig. 1, the performance of ordinary AD methods drops a lot when the training set contains anomalies. Previous works such as SoftPatch [12] and InReaCh [14] have noticed this problem, and they try to remove erroneous abnormal features while building the memory bank. However, They have weaker capabilities for logical global anomalies. In this paper, we resolve this problem from the perspective of model training for distillation-based AD methods.

We propose AD^2, to achieve **A**nomaly **D**etection during training an **A**nomaly **D**etection model. AD^2 can reduce the negative impact of possible abnormal

samples on model training and enables AD methods to automatically exploit uncertain normal data in a fully unsupervised manner.

Different from some pioneers that optimize the nearest neighbor selection logic, AD^2 is developed based on a distillation-based framework. We focus on how to reduce the negative effects of abnormal samples or features during model training, making the model learning process more robust. Distillation-based AD methods cultivate the student models to learn the imitation ability from the teacher model, which is problematic under noisy conditions, e.g. anomalous samples will mislead the student model to imitate anomalies. AD^2 follows the principles that the student model should imitate a majority of training samples containing consistent characteristics, and ignore the learning of rare prominent samples. The model is concentrated on the major feature discrepancy learning so that anomaly detection and rejection are achieved in the training phase. To the best of our knowledge, we are the first to resolve noisy AD tasks by optimizing model training and being valid on logical anomalies.

The proposed AD^2 contains a Non-Major Feature Elimination (NMFE) module and an Anomaly Training Data Removal (ATDR) strategy. The NMFE module reduces the feature learning gradient of occasional differences according to the mutual information between training samples in a batch, ensuring that the parameter optimization occurs only in consistent responses. The ATDR strategy evaluates the feature of training data at current step and suppresses the training weight of outliers to reduce the negative impact of potential anomalous samples. Experiments show that training data mixed with abnormal samples deteriorates the performance of ordinary AD methods. In this case, AD^2 can make the anomalous samples be processed properly and alleviate the performance deterioration. Meanwhile, the inference efficiency remains the same as before.

The contributions can be summarized as follows:

- We propose a novel approach to tackle the problem of normal samples being noisy, which could prevent the performance of AD methods from deteriorating due to the potential defective samples in the training set. The proposed AD^2 can deal with more practical data, and achieves a fully unsupervised procedure for AD tasks.
- The AD^2 adopts a proposed NMFE module to rely on the mutual information for more consistent feature learning, reducing adverse effects from anomalous samples. It also uses the proposed ATDR strategy to remove potential abnormal samples from participating in training, promoting model training in a cleaner context. AD^2 makes the model less sensitive to the presence of noisy data.
- The experiments on the three standard benchmarks demonstrate the effectiveness of the proposed method. In different scenarios and different ratios of noise, AD^2 can significantly alleviate the performance degradation introduced by noisy training data.

2 Related Work

2.1 Anomaly Detection Methods

Recent AD methods for image anomaly detection can be roughly categorized into reconstruction-based, memory-based, and distillation-based [10,13,23,26].

Reconstruction-based methods [15,17,28,29] suppose that the anomalies cannot be reconstructed well because they do not appear in the training set and the model did not learn the corresponding ability [19]. However, reconstructed-based methods face a risk of generating anomaly areas accurately due to the over-generalization issue [17]. Memory-based methods [1,6,8,27] usually build a memory bank of normal features during training [31] and calculate the distance for a query image during testing. Distillation-based methods [9,20,21] are based on the idea of boundaries of education. The student model is trained on an anomaly-free dataset and is expected to mimic the behavior of the teacher model [3]. For abnormal samples, The output by the student model and teacher model should be differentiated. GCAD [3] extends this thought to a global scope to detect logical anomalies. Distillation-based methods are efficient and practical in real scenarios. Recently EfficientAD [2] achieves accurate detection at a breakneck speed.

However, the AD methods mentioned above assume the training data is clean, which may encounter problems in practical scenarios. We break unrealistic clean assumptions and explore scenarios with noisy training data.

2.2 Anomaly Detection Methods with Noise

Several works have noticed that polluted training data would affect the performance of anomaly detection models. SROC [7] proposed a simple refinement strategy to filter the polluted images. It used ensembles of classifiers trained on different splits of the training data and then removed training samples with high anomaly scores [7]. SoftPatch [12] optimized the memory bank building procedure of PatchCore [18]. PatchCluster [30] defined this setting as blind AD, and it proposed to resolve the outlier among patches. InReaCh [14] assumed that normal patches should be well associated across training images. It associated patches into channels and selected channels with high confidence.

Above mentioned methods optimized the feature nearest neighbor selection by refining the memory bank, using pre-defined measurements to find the possible outliers, and excluding the patch feature. From another perspective, our proposed AD^2 is based on a distillation framework and focuses on model training. We release the ability of the AD methods to both training and testing stages, making anomaly detection supervised by anomaly detection.

3 Anomaly Detection During Training

3.1 Task Formulation

This section formally describes the problem that we aim to solve. We define the anomaly detection tasks in a formal formula. Under desired conditions, a training dataset $D_{train}^{N_1} = \{x_1, ...x_{N_1}\}$ consisting of normal images is given during the training stage, $y(x_i)$ is the label of $x_i \in D_{train}^{N_1}$, and $y(x_i) = 0$. N_1 is the count of normal training samples. For input image $q_i \in D_{test}^{M} = \{q_1, q_2, ..., q_M\}$ during the testing stage, the label of q_i is $y(q_i) \in \{0, 1\}$ can be normal or abnormal. The AD method φ should capture information from $D_{train}^{N_1}$ and distinguish q_i. We expect that the value of $\varphi(q_i) = \{0, 1\}$ consistent with $y(q_i)$, declaring q_i is normal or abnormal correctly.

Unfortunately, in practical scenarios, the training dataset usually contains some potentially anomalous samples. A real-world training dataset would be $D_{train}^{N_1+N_2} = \{x_1, ...x_{N_1}, x_{N_1+1}, ..., x_{N_1+N_2}\}$, in which $y(x_i) \in \{0, 1\}$. The $x_i \in \{x_{N_1+1}, ..., x_{N_1+N_2}\}$ is an abnormal sample and is the count of undesired samples. The abnormal samples mixed into the training set will break the distribution boundary of the $D_{train}^{N_1}$, causing trouble to the modeling procedure. Our proposed method considers this case and deals with $D_{train}^{N_1+N_2}$ properly to maintain the discriminating ability on the test set D_{test}^{M}.

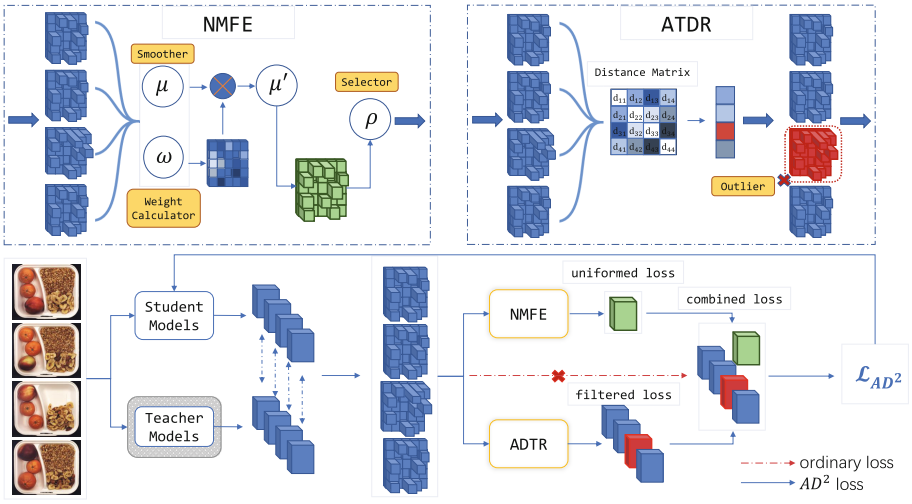

Fig. 2. The illustration of the proposed AD^2 framework. Following the standard paradigm of distillation-based AD methods, the training stage of AD^2 expects the student models can learn the feature extraction ability of the teacher model for normal samples. Different from the ordinary AD methods, AD^2 does not use the discrepancy between student models and teacher models directly. The discrepancy is passed to the NMFE module to get a common innocuous discrepancy information. Meanwhile, the ADTR strategy is applied to remove some potential samples from training. In this way, the training process could be executed in an anomaly detection manner.

3.2 Revisiting Distillation-Based AD Training

Distillation-based AD methods [2,5] focus on modeling learning capabilities for normal data. A pre-trained teacher model with fixed parameters is usually utilized to extract the features of the normal samples. Model training enables the student models to give the same output as the teacher model for the same input image. For unseen abnormal data, the imitation ability of the student models will be limited, so that the output difference between them and the teacher model can be used to estimate the anomaly score.

Compared with reconstruction-based and memory-based methods, distillation-based methods do not require an image reconstruction process, additional storage, and feature retrieval processes, which makes them more efficient. However, they are vulnerable to noisy data. When the training data contains abnormal data, the performance drops dramatically. This shortcoming motivates us to improve the distillation-based methods for the presence of abnormal data.

By digging deeper into the intrinsic logic, we realized that the AD methods can handle potentially abnormal data through two actions. First, the mutual information between one batch should be used during training. The model should only use highly consistent feature discrepancy information to optimize the parameters, which play a role in noise suppression. Besides, the seen data should be used to estimate the proximity of the current samples to others and the training weight of outliers with larger distances should be reduced. These two ideas led to the development of the Non-Major Feature Elimination (NMFE) module and the Anomaly Training Data Removal (ATDR) strategy.

3.3 Non-major Feature Elimination Module

The direct harm caused to the model by mixing defective data is that it provides false normal features. These anomalous samples expand the correct distribution boundaries of normal samples, which may overlap with testing abnormal samples. In order to avoid negative content, we utilize the mutual information between the training samples within a batch, to make the model training more cautious.

For a training sample x_i, the feature extracted by the student model is $\mathcal{M}_S(x_i)$, and the feature extracted by the teacher model is $\mathcal{M}_T(x_i)$. For distillation-based methods, the training procedure is designed to minimize the discrepancy between $\mathcal{M}_S(x_i)$ and $\mathcal{M}_T(x_i)$. We define the discrepancy as $\Delta_{ST}(x_i) = ||\mathcal{M}_S(x_i) - \mathcal{M}_T(x_i)||_2$. Suppose $\chi = \{x_i | i = 1, ..., b\}$ is the training samples of a batch, where b is the batch size hyper-parameter. In the conventional training procedure, $\Delta_{ST}(x_i)$ is used to optimize the model parameters, and then the loss for a batch is $\mathcal{L} = \Delta_{ST}(\chi)$. In our case, we propose a Non-Major Feature Elimination (NMFE) module to filter out the atypical feature discrepancy within a batch. The NMFE module refines the feature discrepancy and then presents $\mathcal{L}_{NMFE} = \texttt{NMFE}(\Delta_{ST}(\chi))$ for better model optimization guidance.

NMFE module consists of three parts: smoother $\mu(\cdot)$, weight calculator $\omega(\cdot)$, and selector $\rho(\cdot)$. We describe each component of them as follows.

Smoother $\mu(\cdot)$ is an averaging operation on the batch axis. It is used to smooth out prominent differences in specific areas of rare samples. Since $\Delta_{ST}(\chi) \in \mathbb{R}^{b\times C\times W\times H}$ represents the feature discrepancy of the training samples of one batch, where C is the channel size, W and H are the size of the feature discrepancy. $\mu(\Delta_{ST}(\chi)) \in \mathbb{R}^{C\times W\times H}$ calculates the mean values across the batch:

$$\mu(\Delta_{ST}(\chi)) = \frac{1}{b}\sum_{i=1}^{b}\Delta_{ST}(x_i) \tag{1}$$

Weight calculator $\omega(\cdot)$ calculates the element-wise weight information, indicating the certainty of each element in the feature discrepancy. $\omega(\Delta_{ST}(\chi)) \in \mathbb{R}^{C\times W\times H}$ measures the certainty of $\mu(\Delta_{ST}(\chi))$ by using the variance information within the batch as the reference. We use a factor $p_{large} \in [0, 1]$ to compute the p_{large}-quantile of the elements of $\sigma(\Delta_{ST}(\chi))$ as σ_{large}. Elements in $\sigma(\Delta_{ST}(\chi))$ which are larger than σ_{large} represent the most inconsistent part of it in the batch dimension. We assign a re-weight factor $\omega_{large} = 0.1$ to reduce the impact:

$$\omega_{c,w,h} = \begin{cases} \omega_{large} & \text{if } \omega_{c,w,h} \geq \sigma_{large} \\ 1.0 & \text{otherwise} \end{cases} \tag{2}$$

Finally, $\rho(\cdot)$ is a hard-example selection operation, used to select the most hard feature discrepancy. We follow [2] to use $\rho(\cdot)$ as a crucial technology. Based on a parameter $p_{hard} \in [0, 1]$, the p_{hard}-quantile of the elements of $\Delta_{ST}(\chi)$ is calculated as d_{hard}, and the elements $f_{i,c,w,h}$ of $\Delta_{ST}(\chi)$ that are larger than d_{hard} are selected:

$$\rho(\Delta_{ST}(\chi)) = \{f_{i,c,w,h} | f_{i,c,w,h} \in \Delta_{ST}(\chi) \geq d_{hard}\} \tag{3}$$

In the original version, it is used to pick the most relevant parts of an image. However, when the training sample is abnormal, directly applying it may introduce anomalous regions. We change its content by encapsulating the input from $\Delta_{ST}(\chi)$ to $\omega(\Delta_{ST}(\chi))\cdot\mu(\Delta_{ST}(\chi))$ so that it will select the most common feature discrepancy and suppress some occasional abnormal differences.

Combining the Eq. (1), Eq. (2), and Eq. (3) the $\mathcal{L}_{NMFE}$ is defined as:

$$\begin{aligned} \mathcal{L}_{NMFE} &= NMFE(\Delta_{ST}(\chi)) \\ &= \rho(\omega(\Delta_{ST}(\chi)) \cdot \mu(\Delta_{ST}(\chi))) \end{aligned} \tag{4}$$

For distillation-based AD methods, the Δ_{ST} reflects the difference between the student model and the teacher model at each step. For the same student models in the same state, the response of the training samples in a batch $\chi = \{x_1, ..., x_b\}$ should be similar. If there exists a certain $\Delta_{ST}(x_i)$ that is particularly prominent in a certain direction than others, implying that the sample x_i is different from other samples. To avoid generating unnecessary gradients for this part of the information, the NMFE module refines the content of $\mathcal{L}$ to $\mathcal{L}_{NMFE}$, with the most common style to execute the gradient back-propagation.

3.4 Abnormal Training Data Removal Strategy

Anomalous samples participating in training can do more harm than good. In addition to limiting overly prominent feature discrepancies by the NMFE module, we also design an Anomaly Training Data Removal (ATDR) strategy to remove potential anomalous samples during training.

The ATDR strategy is executed from the batch scope. For training samples in one batch, their mutual distance matrix is calculated based on their feature map. The element that differs most from other elements in the batch will be temporarily removed from this iteration of training.

$$d(\chi) = \{\sum_{j=1}^{b} dist(x_i, x_j), j \neq i | i = 1, ..., b\} \tag{5}$$

$$Outlier(\chi) = \{x_k | k = argmax(d(\chi))\} \tag{6}$$

Here *dist* is the function to calculate the distance based on the features of two samples. The feature maps are calculated by the student models. After identifying the outlier, the elements of the current training batch are redefined as:

$$\tilde{\chi} = \{x_i \in \chi\} - Outlier(\chi) \tag{7}$$

We reduce the training weights ξ of $Outlier(\chi)$. Then the loss is changed to:

$$\mathcal{L}_{ATDR} = \Delta_{ST}(\tilde{\chi}) + \xi * \Delta_{ST}(Outlier(\chi)) \tag{8}$$

The ATDR strategy prevents the impact of a small number of abnormal samples from model training. Besides, it allows the training procedure to be carried out following the conventions of curriculum learning.

3.5 The Overall Process

The overall process of AD^2 is depicted in Fig. 2. The framework follows a standard paradigm of distillation-based AD methods, except that the outputs from student models and teacher models are not used to calculate the loss directly. The features are passed into the NMFE module and the ATDR strategy to obtain innocuous discrepancy loss information as $\mathcal{L}_{AD^2} = \mathcal{L}_{NMFE} + \mathcal{L}_{ATDR}$.

4 Experiments

Table 1. Image-level AUROC performance on three datasets injected with 10% anomalous samples. LOCO means the MVTec LOCO dataset and its structural and logical evaluation results are reported in "LOCO S." and "LOCO L." respectively. The overall mean value is calculated by averaging the values of MVTec LOCO Mean, MVTec AD Mean, and VisA Mean. Methods with the † symbol are noise-aware AD methods, and others are ordinary AD methods. The best indexes are marked in **bold.**

Method ↓ Dataset →	LOCO S.	LOCO L.	LOCO Mean	MVTec AD	VisA	Mean	latency[ms]
PatchCore [18]	–	–	0.579	0.682	0.349	0.537	32
EfficientAD-M [2]	0.864	0.721	0.793	0.904	0.917	0.871	5
SoftPatch [12] †	–	–	0.672	**0.982**	0.910	0.855	32
InReaCh [14] †	–	–	0.753	0.901	0.792	0.815	41
AD^2 †	**0.921**	**0.809**	**0.865**	0.947	**0.946**	**0.919**	5

Table 2. Image-level AUROC performance on the MVTec LOCO dataset injected with 10% anomalous samples. Methods with the † symbol are noise-aware AD methods, and others are ordinary AD methods. The best indexes are marked in **bold.**

Method ↓ Category →	breakfast box	juice bottle	pushpins	screw bag	splicing connectors	mean
PatchCore [18]	0.550	0.707	0.531	0.562	0.543	0.579
EfficientAD-M [2]	0.765	0.956	0.785	0.614	0.843	0.793
SoftPatch [12] †	0.707	0.833	0.617	0.599	0.603	0.672
InReaCh [14] †	0.685	0.931	0.716	0.680	0.751	0.753
AD^2 †	**0.842**	**0.983**	**0.876**	**0.680**	**0.942**	**0.865**

4.1 Experimental Setup

Datasets. We use MVTec LOCO [3], MVTec AD [4], and VisA [32] as the benchmarks. The MVTec AD dataset contains 15 different sub-datasets. The MVTec LOCO dataset contains 5 sub-datasets, each containing situations about the structure anomalies and logical anomalies. The VisA dataset is one of the largest industrial anomaly detection datasets containing 12 sub-datasets. To simulate the real-world scene where the training data contains anomalies, we pollute the original training data according to the different noise ratio settings. Specifically, we collect all the anomalous samples from the test set and randomly select a certain amount of them to inject into the training set. When the pollution procedure is finished, the noisy training set is larger than the original one and intersects with part of the test set, which is also the *Overlap* setting mentioned in [12].

Evaluation Metric. Following the previous works [12], the performance of AD methods is evaluated by calculating the area under the receiver operating characteristics (AUROC). The image-level AUROC for each category is reported.

Implementation Details. We use PyTorch [16] as the basic framework. The input images are resized to 256×256. The initial learning rate is set to 0.0001. We set Adam with weight decay equals $1e-5$ as the optimizer. We enlarge the batch size to 8 to facilitate the utilization of mutual information between training samples. The training iteration is set to 10000. The other experimental settings follow [2]. In the NMFE module, we set ω_{large} to 0.1. In the ATDR strategy, we set ξ to 0.1. The p_{large} and p_{hard} are set to 0.999 following the EfficientAD [2].

4.2 Performance in Noisy Scenarios

To verify the ability of proposed AD^2 , we compare the performance of different methods in noisy scenarios, e.g., polluted datasets. A certain proportion of abnormal data is injected into the training set of the original datasets.

Experiments on Three Datasets. The overall performance on MVTec LOCO, MVTec AD, and VisA datasets with the ratio of noise to 0.1 is reported in Table 1. We can find that in a noisy scenario, common outstanding AD methods like PatchCore [18] and EfficientAD-M [2] have a certain degree of performance degradation. For example, EfficientAD-M achieves 0.907 image-level AUROC on the original MVTec LOCO dataset and only obtains 0.793 in this case, with a drop around 10%. Ordinary AD methods make an unrealistic assumption that the training data is clean and therefore they obtain lower performance. Distillation-based methods achieve higher inference speed and better performance on logical anomalies scenarios. Although noise-aware AD methods like SoftPatch [12] can restore some accuracy (especially on the MVTec AD dataset), they require larger storage space and higher computational cost, which is unrealistic in practical. The proposed AD^2 can effectively deal with the complex noisy situation, and achieve better performance efficiently.

Experiments on the MVTec LOCO Dataset. The experimental results on the MVTec LOCO dataset are reported in detail in Table 2. This dataset contains various logical anomalies, so patch-level nearest neighbor optimization is not suitable for solving this problem. PatchCore only archives 0.579, which is far away from its performance on the MVTec AD dataset. Original EfficientAD-M also drops around 10% performance in this noisy situation. Benefiting from the NMFE module that resolves the anomalous samples from a global perspective, it effectively facilitates AD^2 to restore in this noisy scenario. Take the *splicing connectors* category as an example, SoftPatch fails to restore the accuracy due to its complex logical anomalous types, but AD^2 obtains an acceptable restoration.

Table 3. The performance of image/pixel-level AUROC is reported on three datasets injected with 10% anomalous samples. The symbol ✓ means the configuration takes effect and the symbol − means not. The last line with NMFE and ATDR checked equals the standard AD^2 solution.

NMFE	ATDR	MVTec LOCO	MVTec AD	VisA
–	–	0.793/0.826	0.904/0.826	0.917/**0.870**
✓	–	0.845/0.844	0.923/0.838	0.940/0.863
–	✓	0.838/0.844	0.924/0.837	0.919/0.868
✓	✓	**0.865/0.845**	**0.947/0.848**	**0.946**/0.862

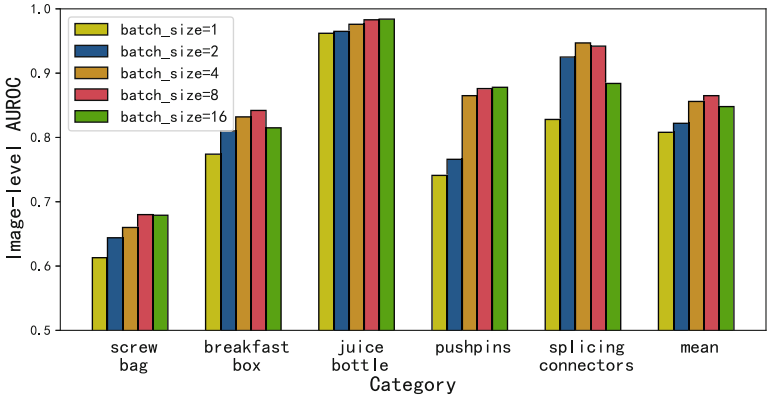

Fig. 3. Exploratory experiments of different batch size settings on MVTec LOCO dataset injected with 10% anomalous samples. The x-axis corresponds to categories and the y-axis is the Image-level AUROC performance. The bar group in each category represents the results of different batch size settings.

4.3 Ablation Studies

Effects of NMFE Module and ATDR Strategy. We conduct ablation studies for the NMFE module and ATDR strategy. The experimental results are reported in Table 3. When neither of them is added, the method falls back to the ordinary AD method, which encounters performance degradation in noisy scenarios. When the NMFE module is applied, the performance obtains obvious improvement, especially in the MVTec LOCO and VisA datasets. ATDR strategy also has a positive effect to some degree. When the two strategies are combined together, the effect of abnormal noise suppression is reflected more clearly.

Table 4. Image-level AUROC performance on MVTec LOCO datasets of different ratios of noise. The performance of EfficientAD-M and proposed AD^2 is compared under different noise ratios on different categories. The Gap ↑ row indicates the improvement or deterioration of AD^2 compared to EfficientAD-M under the same ratio of noise.

Noise Ratio	Method	breakfast box	juice bottle	pushpins	screw bag	splicing connectors	mean
0.0	EfficientAD-M	0.869	0.990	0.969	0.737	0.970	0.907
	AD^2	0.875	0.983	0.960	0.724	0.973	0.903
	Gap ↑	+0.006	−0.007	−0.009	−0.013	+0.003	−0.004
0.1	EfficientAD-M	0.765	0.956	0.785	0.614	0.843	0.793
	AD^2	0.842	0.983	0.876	0.680	0.942	0.865
	Gap ↑	+0.077	+0.027	+0.091	+0.066	+0.099	+0.072
0.2	EfficientAD-M	0.641	0.929	0.659	0.511	0.772	0.702
	AD^2	0.759	0.951	0.761	0.639	0.892	0.800
	Gap ↑	+0.118	+0.022	+0.118	+0.128	+0.120	+0.098
0.3	EfficientAD-M	0.631	0.879	0.588	0.369	0.709	0.635
	AD^2	0.725	0.931	0.751	0.588	0.849	0.769
	Gap ↑	+0.094	+0.052	+0.163	+0.219	+0.140	+0.134
0.4	EfficientAD-M	0.529	0.868	0.475	0.322	0.664	0.572
	AD^2	0.709	0.913	0.634	0.542	0.846	0.729
	Gap ↑	+0.180	+0.045	+0.159	+0.220	+0.182	+0.157

Effects on Different Batch Sizes. The advantage of AD^2 compared to ordinary AD methods is that it uses the mutual information of samples in a batch to identify potential abnormal samples. The hyper-parameter batch size directly affects the improvement. We conduct experiments to verify the suspect, and effects on different batch size can be viewed in Fig. 3. Experiments show that compared to one sample per batch, proper batch size can be utilized obtain positive benefits and it is not sensitive.

4.4 Effects on Different Noise Ratio

The number of abnormal samples mixed into the training data may vary on different scenario. We conduct experiments to verify the effectiveness of the proposed AD^2 at different noise ratios. We generate noisy MVTec LOCO datasets according to the rules described in Sect. 4.1 with different ratio settings. The results can be viewed in Table 4. We can find that as the ratio of noise increases, the performance of AD methods continues to decline, which conforms to intuition. Compared to EfficientAD-M, AD^2 can effectively alleviate the negative impact by the noisy data and restore pronounced performance. From all categories, AD^2 can achieve higher indicators in noisy environments, proving its robustness and versatility.

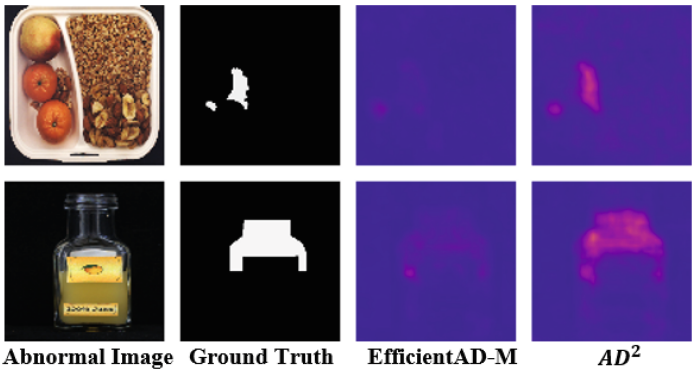

Fig. 4. Visualization of examples and anomaly scores on the MVTec LOCO dataset for different AD Methods. The first and second column represent the anomalous samples and their corresponding ground-truth masks. The 3th and 4th columns are the heatmaps of the anomaly score generated by EfficientAD-M and the AD^2 method.

4.5 Anomaly Score Visualization

We visualize the anomaly score generated by the EfficientAD-M and the AD^2 method in Fig. 4. The models are trained on the polluted training set of the MVTec LOCO dataset and the images are randomly selected from the test set. As can be seen from the 3th column, the anomaly score from the EfficientAD model seems to be less clear and has insufficient differentiation between normal and abnormal areas. The 4th column generated by the AD^2 method obviously improves the shortcomings and can distinguish abnormal areas more clearly.

5 Conclusion

Existing AD methods usually assume that the training data is immaculate, which is often difficult to achieve in practice. Our proposed method AD^2 can introduce anomaly detection to the training process of an AD model, alleviating the impact of noisy data on performance. We propose the NMFE module to adopt the mutual information between samples, eliminating unnecessary disturbance. The ATDR strategy is designed as the guidance for the model training, which deals with the uncertain samples properly. Extensive experiments show that ordinary AD methods deteriorate when the training data are mixed with abnormal samples, while our AD^2 can effectively suppress noise interference and achieve robust anomaly detection. We hope the proposed method can provide inspiration for later research, and be helpful for practical applications, especially in reducing the cost of human labor in data pre-processing.

References

1. Bae, J., Lee, J.H., Kim, S.: PNI: industrial anomaly detection using position and neighborhood information. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 6373–6383 (2023)
2. Batzner, K., Heckler, L., König, R.: EfficientAD: accurate visual anomaly detection at millisecond-level latencies. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pp. 128–138 (2024)
3. Bergmann, P., Batzner, K., Fauser, M., Sattlegger, D., Steger, C.: Beyond dents and scratches: logical constraints in unsupervised anomaly detection and localization. Int. J. Comput. Vision **130**(4), 947–969 (2022)
4. Bergmann, P., Fauser, M., Sattlegger, D., Steger, C.: MVTec AD–a comprehensive real-world dataset for unsupervised anomaly detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 9592–9600 (2019)
5. Bergmann, P., Fauser, M., Sattlegger, D., Steger, C.: Uninformed students: student-teacher anomaly detection with discriminative latent embeddings. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 4183–4192 (2020)
6. Cohen, N., Hoshen, Y.: Sub-image anomaly detection with deep pyramid correspondences. arXiv preprint arXiv:2005.02357 (2020)
7. Cordier, A., Missaoui, B., Gutierrez, P.: Data refinement for fully unsupervised visual inspection using pre-trained networks. arXiv preprint arXiv:2202.12759 (2022)
8. Defard, T., Setkov, A., Loesch, A., Audigier, R.: PaDiM: a patch distribution modeling framework for anomaly detection and localization. In: Del Bimbo, A., Vezzani, R. (eds.) ICPR 2021. LNCS, vol. 12664, pp. 475–489. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-68799-1_35
9. Deng, H., Li, X.: Anomaly detection via reverse distillation from one-class embedding. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 9737–9746 (2022)
10. Guo, H., et al.: Template-guided hierarchical feature restoration for anomaly detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 6447–6458 (2023)
11. Huang, C., Guan, H., Jiang, A., Zhang, Y., Spratling, M., Wang, Y.F.: Registration based few-shot anomaly detection. In: European Conference on Computer Vision, pp. 303–319. Springer (2022)
12. Jiang, X., et al.: SoftPatch: unsupervised anomaly detection with noisy data. Adv. Neural. Inf. Process. Syst. **35**, 15433–15445 (2022)
13. Liu, J., et al.: Deep industrial image anomaly detection: a survey. arXiv preprint arXiv:2301.11514 (2023). **2**
14. McIntosh, D., Albu, A.B.: Inter-realization channels: unsupervised anomaly detection beyond one-class classification. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 6285–6295 (2023)
15. Mousakhan, A., Brox, T., Tayyub, J.: Anomaly detection with conditioned denoising diffusion models. arXiv preprint arXiv:2305.15956 (2023)
16. Paszke, A., et al.: PyTorch: an imperative style, high-performance deep learning library. Adv. Neural. Inf. Process. Syst. **32**, 8026–8037 (2019)
17. Pirnay, J., Chai, K.: Inpainting transformer for anomaly detection. In: International Conference on Image Analysis and Processing, pp. 394–406. Springer (2022)

18. Roth, K., Pemula, L., Zepeda, J., Schölkopf, B., Brox, T., Gehler, P.: Towards total recall in industrial anomaly detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 14318–14328 (2022)
19. Rudolph, M., Wandt, B., Rosenhahn, B.: Same same but different: semi-supervised defect detection with normalizing flows. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pp. 1907–1916 (2021)
20. Rudolph, M., Wehrbein, T., Rosenhahn, B., Wandt, B.: Asymmetric student-teacher networks for industrial anomaly detection. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pp. 2592–2602 (2023)
21. Salehi, M., Sadjadi, N., Baselizadeh, S., Rohban, M.H., Rabiee, H.R.: Multiresolution knowledge distillation for anomaly detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 14902–14912 (2021)
22. Tao, X., Gong, X., Zhang, X., Yan, S., Adak, C.: Deep learning for unsupervised anomaly localization in industrial images: a survey. IEEE Trans. Instrum. Meas. (2022)
23. Tien, T.D., et al.: Revisiting reverse distillation for anomaly detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 24511–24520 (2023)
24. Wang, Y., Peng, J., Zhang, J., Yi, R., Wang, Y., Wang, C.: Multimodal industrial anomaly detection via hybrid fusion. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 8032–8041 (2023)
25. Wyatt, J., Leach, A., Schmon, S.M., Willcocks, C.G.: AnoDDPM: anomaly detection with denoising diffusion probabilistic models using simplex noise. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 650–656 (2022)
26. Yao, X., Li, R., Zhang, J., Sun, J., Zhang, C.: Explicit boundary guided semi-push-pull contrastive learning for supervised anomaly detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 24490–24499 (2023)
27. Yi, J., Yoon, S.: Patch SVDD: patch-level SVDD for anomaly detection and segmentation. In: Proceedings of the Asian Conference on Computer Vision (2020)
28. Zavrtanik, V., Kristan, M., Skočaj, D.: DRAEM-a discriminatively trained reconstruction embedding for surface anomaly detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 8330–8339 (2021)
29. Zavrtanik, V., Kristan, M., Skočaj, D.: DSR–a dual subspace re-projection network for surface anomaly detection. In: European Conference on Computer Vision, pp. 539–554. Springer (2022)
30. Zhang, J., Suganuma, M., Okatani, T.: That's bad: blind anomaly detection by implicit local feature clustering. Mach. Vis. Appl. **35**(2), 31 (2024)
31. Zhang, X., Li, S., Li, X., Huang, P., Shan, J., Chen, T.: DeSTSeg: segmentation guided denoising student-teacher for anomaly detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 3914–3923 (2023)
32. Zou, Y., Jeong, J., Pemula, L., Zhang, D., Dabeer, O.: Spot-the-difference self-supervised pre-training for anomaly detection and segmentation. In: European Conference on Computer Vision, pp. 392–408. Springer (2022)