

FSLC: Fast Scoring with Learnable Coreset for Zero-shot Industrial Anomaly Detection

Songtao Ni

<https://cvl.sjtu.edu.cn/student/sss>

Yuxin Li

<https://cvl.sjtu.edu.cn/student/sss>

Xu Zhao

<https://cvl.sjtu.edu.cn/lead/teacher>

The Computer Vision Lab

Shanghai JiaoTong University

Shanghai, China

Abstract

This paper presents an efficient approach for zero-shot anomaly classification (AC) and segmentation (AS) in industrial applications. While existing zero-shot anomaly detection methods often rely on supplemental prior knowledge or trade computational speed for performance, our method eliminates dependence on external data and accelerates detection. We propose a dynamic coreset strategy that learns directly from test data and repurposes it for anomaly scoring. The coreset is initially constructed from diverse image patches to comprehensively capture potential data patterns across all test samples. Through iterative expansion and score-based filtration, the coreset progressively refines its representation of normal data distributions. This adaptive process enables quantitative evaluation of anomaly severity based on deviations from the learned norms. Experimental validation across multiple benchmarks demonstrates the method’s effectiveness. On MVTec AD, we achieve state-of-the-art average AUROC scores of 93.52% (AC) and 96.55% (AS), while maintaining processing speeds of 5 to 7 frames per second. These results highlight the ability of our framework to balance accuracy and efficiency in practical industrial deployments.

1 Introduction

Industrial anomaly detection is crucial for quality control, where defective products pose risks and false alarms increase costs [16, 29]. Since defect patterns are unpredictable, unsupervised methods that require no defect-specific training have emerged as practical solutions [25]. These approaches measure deviations from normal patterns in full-shot, few-shot, or zero-shot configurations.

Research on full-shot and few-shot anomaly detection algorithms is relatively mature [25], as a certain quantity of normal images serves as direct references, allowing for the analysis of discrepancies between the test data and these references from multiple perspectives. PatchCore [20] is efficient but requires labeled data. While full/few-shot methods (using labeled normal references) are well-established, zero-shot detection avoids training data

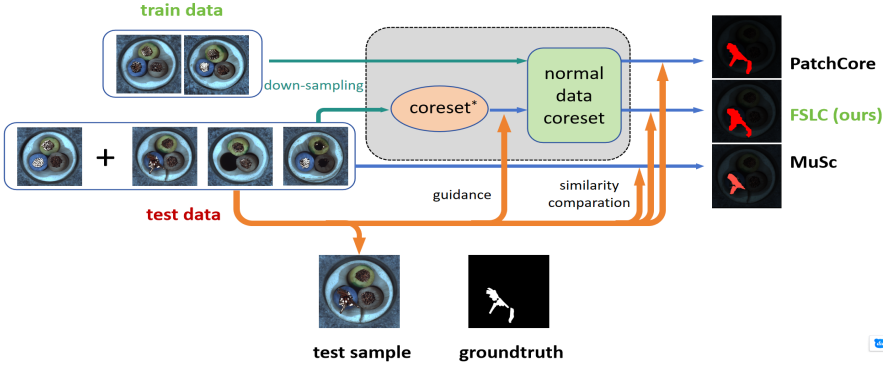


Figure 1: Comparison between our FSLC method and other unsupervised anomaly detection methods. The MuSc[15] uses all test data for anomaly scoring, resulting in high computational complexity and interference from outlier data. Our method introduces the concept of coreset in PatchCore and learns a smaller feature set for scoring. The method achieves state-of-the-art performance in defect localization, and even outperforms the vast majority of unsupervised methods with full train samples like PatchCore[20].

dependency, preventing distribution-shift errors. Current zero-shot research focuses on external data (e.g., pre-trained models), creating impractical dependencies. Though MuSc [15] eliminates this need, its $O(n^2)$ complexity hinders industrial deployment.

To address the computational complexity issues inherent in the MuSc method, our proposed approach integrates the coreset mechanism into the test data processing. In contrast to PatchCore [20] approach of constructing a static coreset, our method introduces a dynamic coreset optimization process as is illustrated in Fig 1. Specifically, we first extract an initial compact coreset* from the unlabeled data. This preliminary coreset then undergoes an iterative refinement process through expansion and filtering operations based on the data distribution. This adaptive approach enables the generation of a more representative coreset that better approximates the true distribution of normal features. The optimized coreset subsequently facilitates efficient re-scoring of all test instances.

Through comprehensive experiments on some datasets such as MVTec AD [11] and VisA [12], our FSLC method demonstrates superior performance across various data scenarios. Both the anomaly classification (AC) and segmentation (AS) capabilities of our algorithm (AC AUROC: 93.52%, AS AUROC: 96.55%) outperform the vast majority of current few-shot and zero-shot methods. Additionally, our detection speed (6.2 frames/s) far exceeds that of the MuSc method (< 0.5 frames/s), which also belongs to the zero-shot category. The contributions of this work are summarized below:

1. We propose an algorithm suitable for zero-shot unsupervised industrial anomaly detection scenarios, which learns knowledge from test data and continuously improves performance as the test data increases.
2. We propose a coreset learning method that relies solely on unlabeled data to realize the expression of normal data.
3. Our algorithm achieves about 15 times speedup over MuSc [15] on datasets like MVTec AD [11] while maintaining comparable AUROC performance.

2 Related Work

In industrial applications, defect detection techniques for both full-shot and few-shot scenarios have reached relative maturity [25], primarily due to the availability of normal data that provides explicit guidance. Building upon full-shot methods, few-shot approaches often maintain robust performance through data augmentation [11] or advanced data utilization techniques [9]. The CutPaste method [13, 23] introduces artificial anomalies to train classification networks, enabling them to learn the boundaries of normal image distributions in feature space. Algorithms like flow-based methods such as FastFlow [28], DiffNet [21], and PyramidFlow [12] employ additional networks to map normal data distributions into Gaussian forms. Reconstruction-based approaches [10, 17, 22] train models with limited data, resulting in constrained reconstruction capabilities that serve as a form of memory. In contrast, methods like PaDiM [8] and PatchSVDD [27] model features at each location as independent Gaussian distributions, proving effective in scenarios with fixed object positions.

The PatchCore method [20] introduces a novel paradigm for anomaly detection by employing coresets to efficiently store representative normal local feature patterns. During inference, the method utilizes k-nearest neighbors (KNN) [6] to compare test image features against the coreset representation, achieving significant reductions in both storage overhead and computational complexity. This coreset-based memory framework has inspired subsequent advancements in the field, particularly evident in few-shot approaches such as WinCLIP+ [9], which integrates CLIP [19] and DINO [2] feature matching mechanisms with coreset-guided anomaly scoring. The coreset concept has been further extended in methods like LTAD [8] and PromptAD [14], while FastRecon [7] incorporates feature space projection and reconstruction techniques. A critical limitation shared by these approaches is their reliance on static coresets, potentially limiting their adaptability to diverse patterns only exist in test data.

In zero-shot anomaly detection scenarios, where access to training data is inherently unavailable, algorithms are required to directly learn representations of normal data from mixed test sets. Contemporary approaches such as WinCLIP [9] and APRIL-GAN [6] leverage pre-defined text prompt, employing the dual encoder architecture of CLIP [5, 19] to establish feature correspondences between image embeddings and textual descriptors. AnomalyCLIP [80] extends this paradigm by integrating supervised learning from domain-specific datasets to refine text prompt optimization. The recently introduced MuSc method [15] presents a novel approach that operates without external datasets or prior knowledge; however, this comes at the cost of significant computational overhead. The method employs a mutual scoring mechanism to evaluate patch-wise similarity and occurrence frequency, establishing high-frequency patterns as statistical indicators of normal regions.

3 Method

3.1 Feature Extraction

We utilize the Vision Transformer (ViT) Φ instead of ResNet [26] for feature extraction. For each image I_i in the test image set $D_{test} = \{I_i, i = 1, \dots, N\}$, we extract intermediate tensors from different layers of the ViT, and concatenate them to represent the multi-scale features of the local regions within the image. Additionally, we apply pooling with a convolution kernel of size $r \times r$ to these multi-scale features, enhancing the receptive field of the features.

Through this method, for each test image I_i , we obtain a map of corresponding patch-level

multi-scale feature vectors $F_i \in \mathbb{R}^{H \times W \times C}$ and an image-level feature vector $\Phi(I_i)$, where $H \times W$ represents the size of the feature map output by the model and C denotes the length of the features. The entire sets of features $\{F_i, i = 1, \dots, N\}$ and $\Phi(D_{test}) = \{\Phi(I_i), i = 1, \dots, N\}$ obtained from the images in the entire test set will collectively be utilized in the subsequent computation process for anomaly detection.

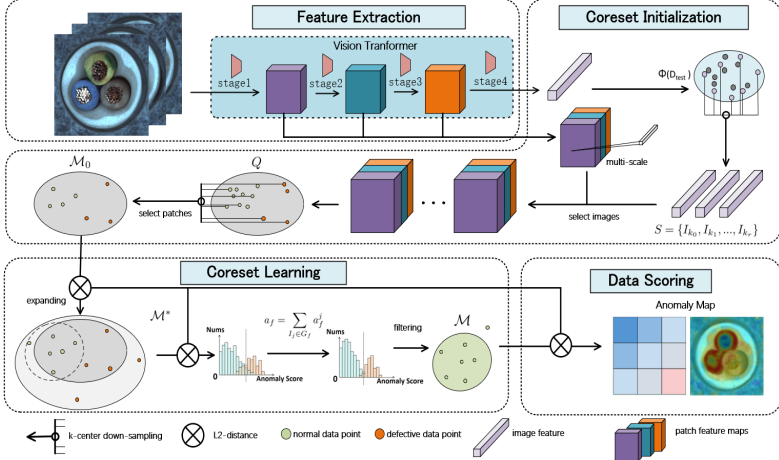


Figure 2: Overview of *FSLC*. Test samples are processed into a series of image-level and patch-level features, and a coreset is formed by the selection of some local features through two rounds of sparse sampling at the image and pixel levels. Expanding and filtering by all patch features allow coreset to include only as many normal features as possible, and it can be used as guidance to give anomaly scores.

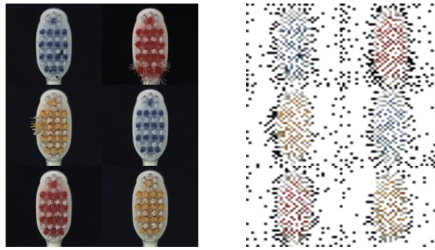


Figure 3: For *toothbrush* category, the left panel presents the image set S . It is evident that the algorithm has selected images of varying styles, enhancing the informational richness of the coreset. The right panel displays the feature set $M_0 \subset Q$. During feature down-sampling, parts with more critical information are more densely sampled.

3.2 Coreset Initialization

After extracting all the features to be tested, we aim to utilize these data to construct an initial coreset M_0 that encapsulates the majority of the patch feature distribution patterns

present.

The strategy adopted in our approach involves firstly selecting as diverse partial images $S = \{I_{k_0}, I_{k_1}, \dots, I_{k_r}\} \subset D_{test}$ as possible under the guidance of image-level features $\Phi(D_{test})$, and subsequently extracting as varied local feature points M_0 from these images based on patch-level features $Q = \{f | f \in F_i, i = k_0, k_1, \dots, k_r\} \in \mathbb{R}^{(r \cdot H \cdot W) \times C}$. r represents the number of images after downsampling at the image-level and F_i is the flattened feature map of the i -th image in S . The process of selecting images or patches essentially constitutes down-sampling. To incorporate guiding information, we opt for selecting feature points as sparsely as possible. Through the k -center greedy [24] method, we can achieve sparse sampling, utilizing an initial coreset M_0 of limited size to delineate the overall data distribution trend within the entire test set.

3.3 Coreset Generation

Due to the presence of many defective patch features mixed within and the absence of certain special data paradigms in the above down-sampling process, the initial coreset M_0 is improper to be directly utilized as a reference for assigning anomaly scores to the test data $\Phi(D_{test})$.

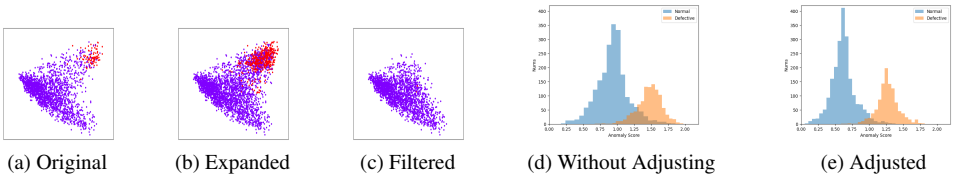


Figure 4: The visualization of the generation process involving the coreset. (a)Initially, the coreset contains a small proportion of anomalous data points, marked in red. (b)The expansion of the coreset enriches the variety of normal data paradigms. (c)The filtering process effectively removes nearly all anomalous data points. (d)(e)In the process of rescoring, adjusting the scores can more effectively separate normal and abnormal points.

Expansion. Among the numerous images $D_{test} - S$ that were not selected, there may still exist a small portion of data paradigms that differ from those already present in the coreset. We evaluate the similarity between data points based on the mutual l_2 distances in the feature space. For each image, we compute the pairwise distances between all its patches and the existing features in the coreset, and select the patches that are far from the data in the coreset within that image. For the feature F_i of the i -th image and existed coreset M_{i-1} , this process can be expressed as: $M_i = M_{i-1} \cup \arg \max_{\Delta M \subset F_i} dist(\Delta M, M_{i-1})$, $dist(\Delta M, M_{i-1}) = \min_{f \in \Delta M, f^* \in M_{i-1}} \|f - f^*\|_2$.

And then we will upgrade coreset from $M_0, M_1, M_2, \dots, M_{N-1}, M_N$. During each iteration, a portion of the features from the i -th image, which were originally not part of the initial coreset, are selectively introduced into the coreset. Following the augmentation process utilizing all features of the test images, the current coreset $M^* = M_N$ now encompasses nearly all data paradigms that may appear in the test data.

Filtering. After obtaining the extended information, coreset M^* can approximately represent all the data paradigms that appear in the test dataset. However, in the process of

selecting as dissimilar data as possible, we inevitably collect a significant amount of feature data corresponding to defective patches. Therefore, we need to filter the current coreset \mathcal{M}^* .

Our algorithm is grounded in a more general and realistic observation: Any normal image patch style will appear across most images—even defective ones— whereas defect-specific patterns are inherently rare. Only when a very limited number of defects happen to cover the corresponding locations fortunately, will a particular normal region be absent. Therefore, we can evaluate the degree of abnormality of patch feature $f \in \mathcal{M}^*$ based on whether similar regions are commonly present in $\Phi(D_{test}) = \{\Phi(I_j) | j = 1, 2, 3, \dots, N\}$:

$$a_f^j = \min_{n=1}^{H \times W} \|f - f_j^n\|_2, a_f^* = \sum_{j=1}^N a_f^j = \sum_{j=1}^N \min_{n=1}^{H \times W} \|f - f_j^n\|_2, \quad (1)$$

Consequently, evaluation scores for the degree of abnormality of data points in the coreset can be obtained using all test images. However, the boundary between the two remains relatively ambiguous. The scores of some normal data points and abnormal ones are close and intermingled. Therefore, for each patch feature f in \mathcal{M}^* , we select the lowest constant value $X\%$ of these scores and compute the average based on this subset $G_f \subset D_{test}$ ($K = N \times X\%$):

$$G_f = \{I_k | a_f^k < a_f^j, I_k \in D_{test}, \forall I_j \in D_{test} - G_f, |G_f| = K\}, \quad (2)$$

$$a_f = \sum_{I_j \in G_f} a_f^j = \sum_{I_j \in G_f} \min_{n=1}^{H \times W} \|f - f_j^n\|_2. \quad (3)$$

As illustrated in Fig 4e, this methodology can reduce the overlap between normal and abnormal features, thereby facilitating a more effective filtration of defective data points in the coreset.

Finally, based on the scoring of the data points in the coreset, we filtered out the portion with higher abnormal scores and obtained the final coreset:

$$\mathcal{M} = \{f | a_f < a_{f^*}, f \in \mathcal{M}^*, \forall f^* \in \mathcal{M}^* - \mathcal{M}, |\mathcal{M}| = N_{wanted}\}. \quad (4)$$

In this module, N_{wanted} denotes the intended size of the coreset that is to be retained. This size can be either manually predetermined, or automatically determined through methodologies such as OTSU [18] method.

3.4 Data Scoring

Through the expansion and filtering of the initial coreset, we obtained a memory bank \mathcal{M} that primarily contains only normal features and encompasses the characteristics of nearly all normal regions that may appear in the test data, which can serve as a reference. Ultimately, the anomaly degree of each patch feature f will be scored based on this reference: $s(f) = \min_{f' \in \mathcal{M}} \|f - f'\|_2$.

After obtaining the anomaly degree score for each patch, the anomaly segmentation(AS) task in defect detection has been achieved. In the other task of anomaly classification(AC), the abnormal score S for each image I_i is calculated using the following formula: $S(I_i) = s(f^*) \cdot w(I_i)$, where $f^* = \arg \max_{f \in I_i} s(f)$, and $w(I_i) = 1 - \frac{\exp(s(f^*))}{\sum_{f \in \mathcal{N}_b(f^*)} \exp(s(f))}$, with $\mathcal{N}_b(f^*)$ the b nearest neighbor points in \mathcal{M} for point f^* . Instead of relying on the single patch with the highest anomaly score, as is commonly done in many algorithms to represent the image-level score, we conduct a analysis of the anomaly scores of several patches with the highest scores within an image.

4 Experiments

4.1 Experiment Settings

Datasets. To rigorously assess our algorithm’s performance in diverse industrial anomaly detection scenarios, we conducted extensive experiments on two benchmark datasets: MVTec AD [10] and VisA [52], which are widely utilized in industrial anomaly detection research.

The MVTec AD dataset consists of high-resolution RGB images across 15 distinct categories, including 10 object categories and 5 texture categories. Each category contains approximately 150 training images exclusively composed of defect-free samples, along with a test set that includes both normal images and various types of defective samples. The VisA dataset, while similar in structure, encompasses 12 different categories. Notably, both datasets provide pixel-level annotations for defect localization in all test images for evaluation.

Evaluation Metrics. For comprehensive quantitative evaluation of our anomaly detection algorithm’s performance, we employ the Area Under the Receiver Operating Characteristic Curve (AUROC) as our primary metric to measure the result of image-level anomaly classification(AC) and pixel-level anomaly segmentation(AS). Compared to classical metrics that require preset threshold, this can adaptively transform the performance within different intervals from different methods to comparable results of the same magnitude.

		MVTec AD		VisA	
		pixel	image	pixel	image
full-shot	PaDiM [9]	97.10	97.90	N/A	N/A
	CutPaste [21]	96.00	95.20	N/A	N/A
	PatchCore(ResNet)[20]	98.10	99.10	N/A	N/A
	PyramidFlow [12]	95.50	N/A	N/A	N/A
	RealNet [45]	99.00	99.60	N/A	N/A
few-shot	PaDiM(k=4) [9]	92.60	80.40	93.20	72.80
	PatchCore(k=4) [20]	94.99	89.49	96.80	85.30
	PatchCore(ViT,k=4)	95.73	92.11	91.36	72.80
	FastRecon(k=4) [9]	96.98	94.24	N/A	N/A
	WinCLIP+(k=4) [9]	96.23	94.65	97.20	87.30
	PromptAD(k=4) [46]	<u>96.50</u>	96.60	97.40	89.10
	WinCLIP [9]	85.12	92.73	79.60	78.10
zero-shot	AnomalyCLIP [43]	91.10	91.50	95.50	82.10
	MuSc(ViT) [45]	97.25	97.83	98.70	92.80
	MuSc(ResNet)	85.51	78.69	82.33	70.75
	FSLC(all)	96.55	93.52	95.55	79.76
	FSLC(part)	<u>95.81</u>	<u>92.33</u>	<u>96.10</u>	75.45

Table 1: Comparison of the anomaly detection performances on both MVTec AD and VisA dataset. We report the average performance of the relevant algorithms tested multiple times on these datasets. Bold indicates the best performance, and horizontal line represents the second best performance.

Baselines. We compare our method with some state-of-the-art approaches on zero-shot or few-shot anomaly detection, *e.g.* WinCLIP [9], AnomalyCLIP [43] and MuSc [45] in Tab 1. In addition, some full-shot methods like PaDiM [9], PatchCore [20], FastRecon [9] and PyramidFlow [12] are also compared.

4.2 Anomaly Detection on MVTec AD and VisA

AUROC performance. Owing to the relatively recent introduction of the VisA dataset, comprehensive evaluations of many full-shot methods remain unavailable. As demonstrated in Tab 3 and Tab 2, we present a comprehensive comparison of anomaly segmentation (AS) and classification (AC) performance between our FSLC method and existing approaches. The

	WinCLIP [9]	AnomalyCLIP [43]	MuSc [45]	FSLC
Candle	88.9 / 95.4	98.8 / 79.3	99.4 / 96.2	98.0 / 87.2
Capsules	81.6 / 85.0	95.0 / 81.5	98.8 / 88.8	94.5 / 75.9
Cashew	84.7 / 92.1	93.8 / 76.3	99.3 / 98.6	95.5 / 75.2
Chewinggum	93.3 / 96.5	99.3 / 97.4	99.5 / 98.3	98.9 / 97.4
Fryum	88.5 / 80.3	94.6 / 93.0	97.8 / 99.0	96.0 / 84.1
Macaroni1	70.9 / 76.2	98.3 / 87.2	99.5 / 89.7	94.4 / 75.1
Macaroni2	59.3 / 63.7	97.6 / 73.4	97.2 / 69.9	84.1 / 48.3
PCB1	61.2 / 73.6	94.1 / 85.4	99.5 / 89.8	98.8 / 55.9
PCB2	71.6 / 51.2	92.4 / 62.2	97.6 / 93.4	93.8 / 71.6
PCB3	85.3 / 73.4	88.4 / 62.7	98.2 / 93.8	96.4 / 81.0
PCB4	94.4 / 79.6	95.7 / 93.9	98.7 / 98.4	96.7 / 88.0
Pipe fryum	75.4 / 69.7	98.2 / 92.4	99.4 / 98.4	99.4 / 96.1

Table 2: FSLC anomaly segmentation and classification performance comparisons in AUROC(%) on VisA dataset in a **pixel-level / image-level** form. We compared FSLC with other zero-shot anomaly detection methods.

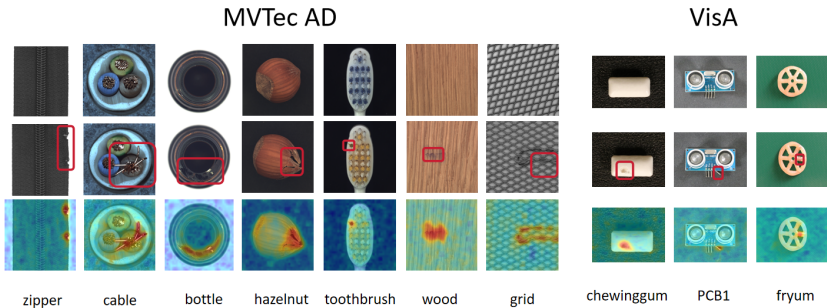


Figure 5: Qualitative results of anomaly localization for both MVTec AD and VisA datasets. The first row presents normal samples in these categories, while the red box in the second row indicates the approximate location of the defect area. The results indicate that our method can effectively locate defect areas by analyzing the distribution trend of test data.

experimental results reveal that our algorithm achieves competitive performance with the state-of-the-art MuSc method across most categories, attaining comparable AUROC performance at both image and pixel levels. Notably, our approach exhibits marginal performance improvements ranging from 0.12% to 0.56% in specific categories. Furthermore, as evidenced in Tab 1, our method substantially outperforms most existing zero-shot and few-shot anomaly detection algorithms, demonstrating its effectiveness across different learning paradigms.

category	PatchCore (k=4) [10]	WinCLIP [10]	WinCLIP+ (k=4) [10]	FastRecon (k=4) [10]	AnomalyCLIP [10]	MuSc [10]	FSLC
Bottle	98.60 / 99.60	89.50 / 99.20	97.80 / 99.30	98.50 / 99.44	90.40 / 89.30	<u>98.60 / 99.90</u>	99.13 / 100.00
Cable	97.90 / 97.40	77.00 / 90.90	94.90 / 88.40	96.12 / 93.79	78.90 / 69.80	<u>96.30 / 99.00</u>	93.88 / 99.74
Capsule	97.70 / 66.30	86.90 / 82.30	96.20 / 77.30	<u>98.96</u> / 90.07	95.80 / <u>87.20</u>	98.90 / 96.70	98.98 / 65.14
Carpet	99.00 / 99.00	95.40 / 100.00	<u>99.30</u> / 100.00	99.15 / 99.90	98.80 / 100.00	99.50 / 99.90	99.20 / <u>99.96</u>
Grid	70.60 / 63.00	82.20 / 98.80	<u>98.00</u> / <u>99.60</u>	86.32 / 88.81	97.30 / 97.00	98.40 / 98.70	<u>98.28</u> / 99.62
Hazelnut	97.00 / 92.80	94.30 / 93.90	<u>98.80</u> / 98.40	98.59 / 99.32	97.10 / 97.20	<u>99.40</u> / <u>99.60</u>	<u>98.46</u> / 99.96
Leather	96.90 / 100.00	96.70 / 100.00	99.30 / 100.00	99.20 / 100.00	98.60 / 99.80	<u>99.70</u> / 100.00	99.82 / 100.00
Metal_Nut	<u>97.00</u> / 94.70	61.00 / 97.10	92.90 / <u>99.50</u>	98.72 / 99.12	74.40 / 93.60	86.00 / 96.30	81.05 / 99.95
Pill	96.90 / 89.00	80.00 / 79.10	97.10 / 92.80	98.32 / 93.48	92.00 / 81.80	97.60 / 96.40	98.16 / <u>94.76</u>
Screw	92.10 / 54.10	89.60 / 83.30	96.00 / 87.90	97.10 / 62.46	<u>97.50</u> / 81.10	98.90 / <u>83.50</u>	96.54 / 57.90
Tile	96.00 / 100.00	77.60 / 100.00	96.60 / 99.90	96.73 / 100.00	94.60 / 100.00	98.10 / 100.00	<u>98.07</u> / 100.00
Toothbrush	98.80 / 95.20	86.90 / 87.50	98.40 / 96.70	99.04 / 93.61	91.90 / 84.70	<u>99.50</u> / 100.00	99.65 / <u>96.94</u>
Transistor	95.00 / <u>98.40</u>	74.70 / 88.00	88.50 / 85.70	94.18 / 97.29	71.00 / 92.80	92.00 / 99.10	<u>94.89</u> / 95.58
Wood	93.10 / 97.40	93.40 / <u>99.40</u>	95.40 / 99.80	94.94 / 99.29	96.50 / 96.80	<u>97.40</u> / 98.50	98.02 / 98.87
Zipper	98.30 / 95.50	91.60 / 91.50	94.20 / 94.50	98.89 / 96.95	91.40 / <u>98.50</u>	<u>98.40</u> / 99.90	94.27 / 94.49

Table 3: FSLC anomaly segmentation and classification performance comparisons in AUROC(%) on MVTec AD dataset in a pixel-level / image-level form. Bold indicates the best performance, and horizontal line represents the second best performance. Our method achieves almost as excellent performance as the PatchCore and MuSc methods.

Time Expense. While achieving comparable AUROC performance, our algorithm demonstrates significantly reduced computational complexity compared to the MuSc. In our experimental evaluation conducted on NVIDIA RTX 4060 GPU using the *PyTorch*, the complete feature processing pipeline—encompassing both coreset generation and anomaly scoring—achieves an average processing time of 30 ms per image. This computational efficiency substantially outperforms most existing methods in terms of runtime performance.


MuSc 	feature extraction	scoring		
time / ms	190.29	1531.08		
FSLC	feature extraction	coreset expanding	coreset filtering	scoring
time / ms	170.41	7.46	6.45	2.44

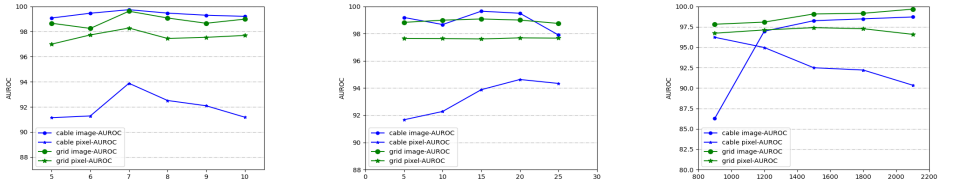
Table 4: Time expense of each image in MuSc and our FSLC method. Tests were conducted on the *leather* category with 83 samples in the MVTec AD dataset, because the average time expense is related to the numbers of test images for MuSc method.

	MVTec		VisA	
	image	pixel	image	pixel
FSLC(ours)	93.52 →	96.55 →	78.95 →	95.91 →
random image sampling	93.13 ↓	92.33 ↓	79.92 ↑	94.12 ↓
random patch sampling	91.16 ↓	83.21 ↓	75.30 ↓	82.23 ↓
w/o coreset expanding	93.88 ↑	96.11 ↓	72.70 ↓	89.89 ↓
w/o score adjusting	90.42 ↓	93.42 ↓	65.60 ↓	93.91 ↓

Table 5: Effects of steps in FSLC. The *random image/patch sampling* refers to using random sampling to replace *k-center greedy* as is illustrated in sec 3.2. Also *w/o coreset expanding* and *w/o score adjusting* refer to removing the modules in sec 3.3 and 3.3. The ↓ represents the decrease in AUROC and ↑ represents the increase.

4.3 Ablation Study

Discussion of the parameters. The selection of some hyper-parameters in the algorithm has a crucial impact on the experimental results. For example, as is illustrated in Fig 6c, the increasing of number of points retained in the final coreset help to enhance the abundance of coreset, but also raise the possibility of mixing incorrect points.



(a) Initial Image Num for Coreset (b) Num of Patches in Expanding (c) Num of Features in Final Coreset

Figure 6: Effects of three hyper-parameters comparisons in anomaly classification and segmentation AUROC(%) on some categories in MVTec AD dataset.

Discussion of the sparse down-sampling and coreset expansion. As evidenced by the experimental results in Tab 5, the performance of sparse down-sampling for both S and \mathcal{M}_0 consistently outperforms random down-sampling. Our findings demonstrate that the combination of sparse down-sampling and coreset expansion effectively maximizes data-representativeness within constrained coreset size, leading to superior detection capabilities.

Discussion of the adjustment of scoring. This approach is specifically designed to optimize the coreset composition by maximizing the retention of normal samples while effectively excluding anomalous data. The experimental results, as shown in Table 5, reveal that employing the unmodified sum of mutual scores significantly degrades AUROC performance.

5 Conclusion

In this work, we present FSLC, a novel zero-shot industrial anomaly detection algorithm that efficiently learns characteristic patterns of normal regions from unlabeled test data. Our

approach operates through three key stages: initial feature extraction and coreset construction, iterative coreset augmentation and filtering using test data for comprehensive representation of normal patterns, and rapid anomaly scoring. The proposed method achieves detection performance comparable to SOTA zero-shot approaches while rivaling the performance of leading full-shot and few-shot methods. Notably, our method demonstrates superior efficiency and broader applicability compared to other zero-shot solutions, offering new choice for practical deployment.

References

- [1] Paul Bergmann, Michael Fauser, David Sattlegger, and Carsten Steger. Mvtec ad—a comprehensive real-world dataset for unsupervised anomaly detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9592–9600, 2019.
- [2] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9650–9660, 2021.
- [3] Xuhai Chen, Yue Han, and Jiangning Zhang. April-gan: A zero-/few-shot anomaly classification and segmentation method for cvpr 2023 vand workshop challenge tracks 1&2: 1st place on zero-shot ad and 4th place on few-shot ad. *arXiv preprint arXiv:2305.17382*, 2023.
- [4] Thomas Defard, Aleksandr Setkov, Angelique Loesch, and Romaric Audigier. Padim: a patch distribution modeling framework for anomaly detection and localization. In *International Conference on Pattern Recognition*, pages 475–489. Springer, 2020.
- [5] 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 *International Conference on Learning Representations*, 2020.
- [6] Eleazar Eskin, Andrew Arnold, Michael Prerau, Leonid Portnoy, and Sal Stolfo. A geometric framework for unsupervised anomaly detection: Detecting intrusions in unlabeled data. *Applications of data mining in computer security*, pages 77–101, 2002. ISSN 1461353211.
- [7] Zheng Fang, Xiaoyang Wang, Haocheng Li, Jiejie Liu, Qiugui Hu, and Jimin Xiao. Fastrecon: Few-shot industrial anomaly detection via fast feature reconstruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 17481–17490, 2023.
- [8] Chih-Hui Ho, Kuan-Chuan Peng, and Nuno Vasconcelos. Long-tailed anomaly detection with learnable class names. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12435–12446, 2024.

- [9] Jongheon Jeong, Yang Zou, Taewan Kim, Dongqing Zhang, Avinash Ravichandran, and Onkar Dabeer. Winclip: Zero-/few-shot anomaly classification and segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19606–19616, 2023.
- [10] Kyosuke Komoto, Shunsuke Nakatsuka, Hiroaki Aizawa, Kunihiro Kato, Hiroyuki Kobayashi, and Kazumi Banno. A performance evaluation of defect detection by using denoising autoencoder generative adversarial networks. In *2018 international workshop on advanced image technology (IWAIT)*, pages 1–4. IEEE, 2018. ISBN 1538626152.
- [11] Mingyu Lee and Jongwon Choi. Text-guided variational image generation for industrial anomaly detection and segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26519–26528, 2024.
- [12] Jiarui Lei, Xiaobo Hu, Yue Wang, and Dong Liu. Pyramidflow: High-resolution defect contrastive localization using pyramid normalizing flow. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14143–14152, 2023.
- [13] Chun-Liang Li, Kihyuk Sohn, Jinsung Yoon, and Tomas Pfister. Cutpaste: Self-supervised learning for anomaly detection and localization. In *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*, pages 9664–9674, 2021.
- [14] Xiaofan Li, Zhizhong Zhang, Xin Tan, Chengwei Chen, Yanyun Qu, Yuan Xie, and Lizhuang Ma. Promptad: Learning prompts with only normal samples for few-shot anomaly detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16838–16848, 2024.
- [15] Xurui Li, Ziming Huang, Feng Xue, and Yu Zhou. Musc:zero-shot industrial anomaly classification and segmentation with mutual scoring of the unlabeled images. In *The 12-th International Conference on Learning Representations*, 2024.
- [16] Takashi Matsubara, Ryosuke Tachibana, and Kuniaki Uehara. Anomaly machine component detection by deep generative model with unregularized score. In *2018 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2018. ISBN 1509060146.
- [17] Takashi Matsubara, Kazuki Sato, Kenta Hama, Ryosuke Tachibana, and Kuniaki Uehara. Deep generative model using unregularized score for anomaly detection with heterogeneous complexity. *IEEE Transactions on Cybernetics*, 52(6):5161–5173, 2020. ISSN 2168-2267.
- [18] Nobuyuki Otsu et al. A threshold selection method from gray-level histograms. *Automatica*, 11(285-296):23–27, 1975.
- [19] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision, 2021. URL <https://proceedings.mlr.press/v139/radford21a.html>.

- [20] Karsten Roth, Latha Pemula, Joaquin Zepeda, Bernhard Schölkopf, Thomas Brox, and Peter Gehler. Towards total recall in industrial anomaly detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14318–14328, 2022.
- [21] Marco Rudolph, Bastian Wandt, and Bodo Rosenhahn. Same same but different: Semi-supervised defect detection with normalizing flows. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 1907–1916, 2021.
- [22] Thomas Schlegl, Philipp Seeböck, Sebastian M Waldstein, Georg Langs, and Ursula Schmidt-Erfurth. f-anogan: Fast unsupervised anomaly detection with generative adversarial networks. *Medical image analysis*, 54:30–44, 2019. ISSN 1361-8415.
- [23] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017.
- [24] Ozan Sener and Silvio Savarese. Active learning for convolutional neural networks: A core-set approach. *arXiv preprint arXiv:1708.00489*, 2017.
- [25] Xian Tao, Xinyi Gong, Xin Zhang, Shaohua Yan, and Chandranath Adak. Deep learning for unsupervised anomaly localization in industrial images: A survey. *IEEE Transactions on Instrumentation and Measurement*, 71:1–21, 2022.
- [26] Sasha Targ, Diogo Almeida, and Kevin Lyman. Resnet in resnet: Generalizing residual architectures. *arXiv preprint arXiv:1603.08029*, 2016.
- [27] Jihun Yi and Sungroh Yoon. Patch svdd: Patch-level svdd for anomaly detection and segmentation. In *Proceedings of the Asian conference on computer vision*, 2020.
- [28] Jiawei Yu, Ye Zheng, Xiang Wang, Wei Li, Yushuang Wu, Rui Zhao, and Liwei Wu. Fastflow: Unsupervised anomaly detection and localization via 2d normalizing flows. *arXiv preprint arXiv:2111.07677*, 2021.
- [29] Jianpeng Zhang, Yutong Xie, Guansong Pang, Zhibin Liao, Johan Verjans, Wenxing Li, Zongji Sun, Jian He, Yi Li, and Chunhua Shen. Viral pneumonia screening on chest x-rays using confidence-aware anomaly detection. *IEEE transactions on medical imaging*, 40(3):879–890, 2020. ISSN 0278-0062.
- [30] Ximiao Zhang, Min Xu, and Xiuzhuang Zhou. Realnet: A feature selection network with realistic synthetic anomaly for anomaly detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16699–16708, 2024.
- [31] Qihang Zhou, Guansong Pang, Yu Tian, Shibo He, and Jiming Chen. Anomaly-clip: Object-agnostic prompt learning for zero-shot anomaly detection. *arXiv preprint arXiv:2310.18961*, 2023.
- [32] Yang Zou, Jongheon Jeong, Latha Pemula, Dongqing Zhang, and Onkar Dabeer. Spot-the-difference self-supervised pre-training for anomaly detection and segmentation. In *European Conference on Computer Vision*, pages 392–408. Springer, 2022.