International Journal of Scientific Engineering and Research (IJSER)
ISSN (Online): 2347-3878
SJIF (2025): 8.036

LMF-Core: A Training-Free Coreset for Industrial Anomaly Detection

Hongzhou Liu

North China Electric Power University, School of Control and Computer Engineering, Beijing, 102206, China
Email: *2351493472[at]qq.com*

Abstract: ***Automated visual inspection must detect surface and structural defects while observing only defect-free samples during setup. Training-free memory-bank methods on frozen ImageNet backbones reach high accuracy, but they concatenate features from several backbone layers with equal weight and rely on an iterative greedy coreset whose memory size is not directly controllable. We address both issues with LMF-Core, a training-free framework with two statistics-only modules. Variance-Guided Fusion prunes low-information channels and weights each layer by the inverse variance of its normal features. Random-Projection Coreset replaces greedy selection with a Johnson–Lindenstrauss projection and uniform subsampling, giving a single, directly controllable memory parameter. Anomaly scores are the nearest-neighbour distance to the bank. On MVTec AD, LMF-Core attains 96.3% image-level and 96.8% pixel-level AUROC (mean over five seeds) from an 8.8 MB bank, with no training, labels, or anomalous examples, providing a compact memory footprint suitable for memory-constrained inspection hardware. Baseline figures are quoted from the original publications, whereas all reported LMF-Core results are obtained experimentally.***

Keywords: Industrial anomaly detection, visual inspection, unsupervised anomaly detection, memory bank, feature fusion, random projection, computer vision

1. Introduction

Visual quality inspection is a core step in manufacturing. Machined metal parts, printed circuit boards, textiles, pharmaceutical blisters, and packaged goods must be checked for scratches, dents, contamination, missing parts, and printing errors before they leave the line. Manual inspection is slow, inconsistent, fatigue-prone, and does not scale to modern throughput, so automated inspection based on computer vision is widely deployed 0, [26]. As production lines diversify and product cycles shorten, an inspection system must also be quick to commission, becoming operational after acquiring several hundred normal images, without a labelling campaign or per-product model training.

The central difficulty is data imbalance. A factory can photograph thousands of normal items easily, but real defects are rare, highly varied, and often unknown in advance. A new defect type can appear that was never seen during setup. Framing inspection as ordinary supervised classification is therefore impractical: there are too few defect examples, and a supervised model would fail on unseen defect types [2]. This motivates the *unsupervised anomaly detection (AD)* setting, now dominant in industrial inspection research 0, [26]. The model is shown *only* normal samples during setup and must, at test time, (a) decide whether an image is normal or anomalous (image-level detection) and (b) localise the anomalous region as a pixel map (pixel-level segmentation). Because no defect labels are used, anything that deviates from the learned model of normality is flagged, which naturally covers unseen defect types.

A particularly practical family of methods avoids training a network altogether: it uses a *frozen backbone pretrained on ImageNet* as a generic feature extractor and reasons about normality directly in feature space [3], [4], [5]. SPADE [3] retrieves nearest normal images and compares pixel features; PaDiM [4] fits a per-position Gaussian to normal patch features; and PatchCore [5] stores a *coreset* of normal patch features in a memory bank and scores test patches by distance to the nearest stored feature. PatchCore reaches high accuracy on standard benchmarks without any training, which supports the memory-bank paradigm for rapid factory deployment.

Accuracy, however, is not the only constraint. Inspection often runs on embedded hardware with limited memory and a strict cycle time, and the most accurate memory-bank pipelines carry two avoidable costs:

1) *Undifferentiated features.* Features from different backbone layers and channels are usually concatenated with *equal weight* [4], [5], which assumes every layer and channel is equally informative across very different products. In practice, some channels carry mostly background or illumination, and one layer is often more discriminative than the other for a given category.
2) *Expensive bank construction.* To bound memory, PatchCore applies an iterative *greedy coreset* [5], [6], an offline optimisation loop whose cost grows with the number of normal patches and whose output size is not directly controllable by the operator.

LMF-Core (Lightweight Multi-Layer Fused Coreset) retains the training-free memory-bank design while addressing both costs: it makes the patch features discriminative and the memory bank low-cost and directly controllable, using two coordinated, training-free modules (Fig. 1):

- *Variance-Guided Fusion (VG-Fusion).* Prune the least informative channels and weight each layer by the inverse variance of its normal features, so stable, discriminative dimensions dominate the representation. All statistics come from the normal setup images.
- *Random-Projection Coreset (RP-Coreset).* Replace greedy selection with a Johnson–Lindenstrauss random projection [6] followed by uniform subsampling at a single ratio r, yielding a non-iterative, directly controllable memory–accuracy parameter.

Anomaly scoring is the standard nearest-neighbour distance to the bank, producing a pixel-level map and an image-level decision. The scoring strategy is intentionally kept simple: a local-density calibration of the distance was also evaluated, but it did not improve accuracy on MVTec AD (Section 6) and is excluded from the final method.

Contributions. (1) A training-free framework, LMF-Core, that unifies discriminative feature fusion and a low-cost, directly controllable coreset memory bank for unsupervised industrial AD. (2) Two lightweight, statistics-only modules, each addressing a concrete limitation of equal-weight memory-bank methods, together with the full set-up/inference algorithm and a complexity analysis. (3) A reproducible evaluation protocol with results on MVTec AD [2]: image- and pixel-level AUROC, a module ablation, a sensitivity study, and qualitative localisation. The same protocol is designed to also transfer to VisA [7] and BTAD [8], which we leave to future evaluation.

The remainder of the paper is organised as follows. Section 2 reviews unsupervised industrial AD. Section 3 formalises the problem. Section 4 presents LMF-Core. Section 5 gives the protocol, Section 6 the results and discussion, Section 7 the limitations, and Section 8 concludes.

2. Related Work

Unsupervised industrial AD methods fall into two broad families: *reconstruction-based* and *feature-embedding* 0, [26]. Table 1 positions representative methods by the properties that matter for deployment.

2.1 Reconstruction-based methods

These learn to reproduce normal images and treat large reconstruction error as evidence of an anomaly. Autoencoders and variational autoencoders were early examples, and GAN-based models such as GANomaly [9] learn the normal distribution adversarially. A weakness is that powerful decoders can also reconstruct anomalies, weakening the signal. DRAEM [10] trains on synthetically corrupted images so the network learns to restore normal appearance and segment the corruption, turning reconstruction into a discriminative task; CutPaste [11] similarly fabricates defects by cut-and-paste augmentation. Diffusion models have recently been used both to reconstruct normal appearance and to synthesise realistic anomalies 0. These methods generally require training a network per category, which lengthens commissioning.

2.2 Feature-embedding methods

These compare images in a learned or pretrained feature space. *Teacher–student* methods train a student to mimic a pretrained teacher on normal data; the mismatch on anomalies is the score [12], and reverse distillation [13] improves this with a reversed encoder–decoder student. *Normalizing-flow* methods map normal features to a simple density and flag low-likelihood test features, including DifferNet [14], FastFlow [15], and CFLOW-AD [16]. Recent single-network designs such as SimpleNet [17] and the efficiency-oriented EfficientAD [18] push accuracy and latency further, but still require training.

Most relevant here are *memory-bank* and *statistical* methods on a *frozen* backbone, which need no training. SPADE [3] compares deep pyramid features of retrieved nearest normal images; PaDiM [4] scores patches by Mahalanobis distance to a per-position Gaussian; and PatchCore [5] stores locally aware patch features, reduces the bank with a greedy coreset [6], and scores by nearest-neighbour distance. LMF-Core belongs to this family and differs along the two axes identified in the introduction: it weights and prunes features by their normal-data variance rather than concatenating equally, and it builds the bank by random projection plus subsampling rather than greedy selection.

2.3 Benchmarks and other settings

MVTec AD [2] is the standard benchmark: 15 categories of objects and textures, normal-only training, and pixel-accurate masks. VisA [7] is larger and harder (12 categories), and BTAD [8] adds three industrial products with subtle textures. Beyond 2D detection, the field has expanded to *few-/zero-shot* detection with vision–language models such as CLIP [24] (e.g., WinCLIP [19]), to *3D and multimodal* detection on MVTec 3D-AD [20], and to *logical* anomalies that violate a global constraint [1]. We target the mature, directly relevant 2D unsupervised setting and treat the harder settings as future work.

Table 1: Property comparison of representative unsupervised AD methods. "TF" = training-free; "Loc." = pixel localisation; "Adapt. feat." = data-adaptive feature weighting; "Ctrl. mem." = directly controllable memory size.

Method	TF	Loc.	Adapt. feat.	Ctrl. mem.
GANomaly [9]	no	weak	no	n/a
DRAEM [10]	no	yes	no	n/a
RD4AD [13]	no	yes	no	n/a
PaDiM [4]	yes	yes	partial	no
SPADE [3]	yes	yes	no	no
PatchCore [5]	yes	yes	no	partial
LMF-Core (ours)	yes	yes	yes	yes

3. Problem Formulation and Preliminaries

Setting. Let $\mathcal{X}_n = \{x^{(1)}, \dots, x^{(N)}\}$ be a set of normal (defect-free) images available at setup. No anomalous images and no labels are available. At test time we receive an image x and must output an anomaly score map $S \in \mathbb{R}^{H \times W}$ and an image-level score $a \in \mathbb{R}$, such that a threshold τ yields the decision $\mathbb{1}[a > \tau]$ and S localises the defect.

Frozen-backbone features. A backbone ϕ pretrained on ImageNet maps an image to a set of feature maps. We use two intermediate maps $\phi_2(x) \in \mathbb{R}^{C_2 \times H_2 \times W_2}$ and $\phi_3(x) \in \mathbb{R}^{C_3 \times H_3 \times W_3}$. Treating each spatial location as a token gives *patch features*. Memory-bank AD [3], [5] stores patch features of $\mathcal{X}_n$ in a bank $\mathcal{M}$ and scores a test patch by its distance to the nearest element of $\mathcal{M}$.

Greedy coreset. To bound $|\mathcal{M}|$, PatchCore [5] applies the greedy k-center coreset [6], which repeatedly adds the feature farthest from the current selection. This minimises the covering radius but is an iterative $O(|\mathcal{M}|\, N_p)$ procedure over the N_p candidate patches and gives only indirect control of

the final size. LMF-Core keeps the goal (a small, representative bank) but reaches it without iteration.

4. Proposed Framework: LMF-Core

4.1 Overview

LMF-Core has a one-off *setup phase* on $\mathcal{X}_n$ and a per-image *inference phase* (Fig. 1). Two modules act in sequence: VG-Fusion builds a discriminative patch representation; RP-Coreset compresses the normal representation into a small bank $\mathcal{M}$. A test image is then scored by the nearest-neighbour distance of its patches to $\mathcal{M}$. No gradient-based training occurs; the only fitted quantities are channel/layer statistics, the random projection matrix, the bank, and the normal-score statistics, all obtained from $\mathcal{X}_n$.

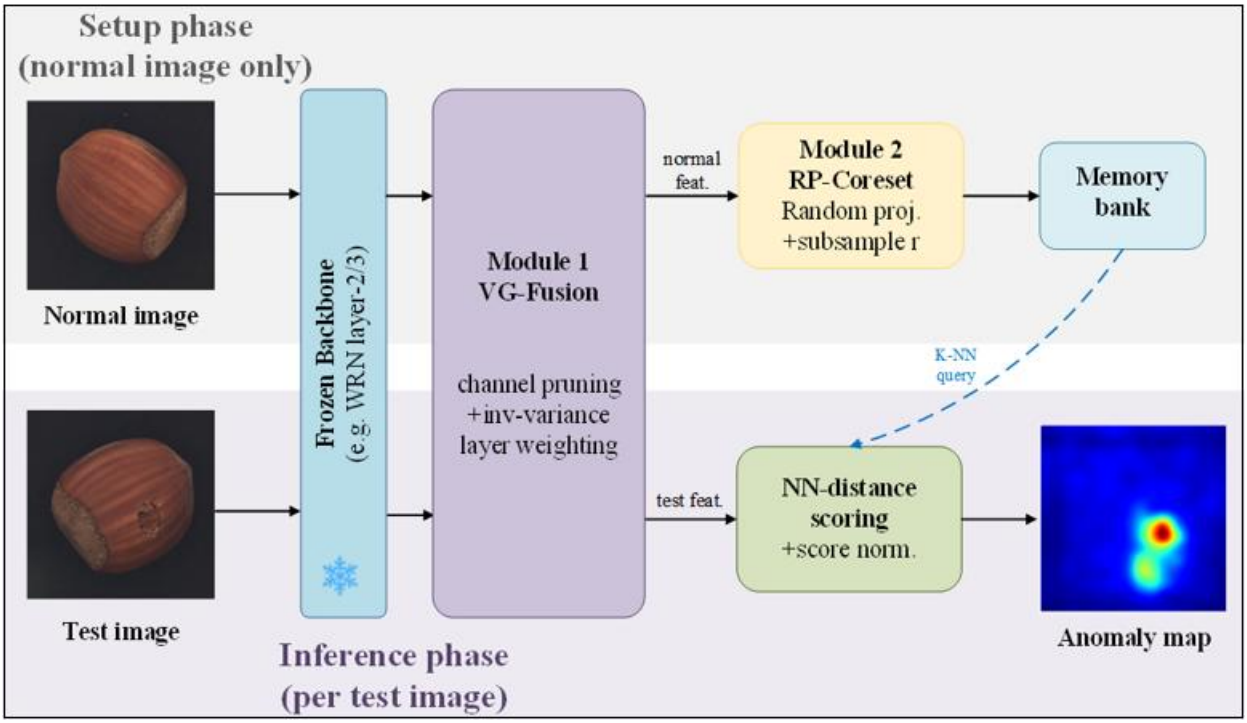

Figure 1: The LMF-Core framework. A frozen ImageNet backbone produces layer-2/layer-3 patch features. *Module 1 (VG-Fusion)* prunes low-information channels and weights the two layers by inverse normal-feature variance to form a discriminative fused feature. *Module 2 (RP-Coreset)* compresses normal features into a small memory bank $\mathcal{M}$ by a random projection and uniform subsampling. At inference, each test patch is scored by its nearest-neighbour distance to $\mathcal{M}$; scores form the anomaly map S and their re-weighted maximum (normalised by the normal-score statistics) is the image score a. The top band is the one-off setup phase (normal images only) and the bottom band the per-image inference phase; the frozen backbone and VG-Fusion are shared by both. The dashed arrow marks the nearest-neighbour query of the bank $\mathcal{M}$ at inference. No training is involved

4.2 Multi-layer patch feature extraction

Let ϕ be a frozen backbone pretrained on ImageNet [21]; we use WideResNet-50 [22], with ResNet variants [23] as drop-in alternatives. We read $\phi_2(x)$ and $\phi_3(x)$: early layers carry generic high-resolution texture, deeper layers carry abstract, lower-resolution structure. We avoid the final layer, whose features are biased toward ImageNet semantics rather than local defects [4], [5]. As the two maps differ in resolution, $\phi_3(x)$ is bilinearly resized to the resolution of $\phi_2(x)$ so patch positions align. Following PatchCore [5], each patch is aggregated with its $p \times p$ neighbourhood ($p = 3$) by adaptive average pooling, adding robustness to small misalignments.

4.3 Module 1: Variance-Guided Fusion

Plain concatenation treats every layer and channel as equally informative. VG-Fusion instead uses two normal-data statistics, computed once and with no labels.

Layer weighting. For layer l, let v_l be the mean over channels of the across-patch variance of normal features. A layer whose normal features cluster tightly (small v_l) makes deviations more detectable and therefore receives a higher weight:

$$w_l = \frac{1/v_l}{\sum_{k\in\{2,3\}} 1/v_k}, \qquad l \in \{2,3\}. \tag{1}$$

Channel pruning. Within each layer, a channel whose normal responses are nearly constant across positions and images carries little discriminative signal (often background or global illumination). Let σ_c be the standard deviation of channel c over all normal patches. We keep the top fraction κ of channels by σ_c and discard the rest, reducing dimension before projection without any learning.

Fused feature. Each retained channel is standardised channel-wise (zero mean, unit variance from normal statistics), each layer is scaled by w_l, and the two layers are concatenated into the fused patch feature $f \in \mathbb{R}^D$. Equal-weight, no-pruning concatenation is the special case $w_2 = w_3 = 0.5$, $\kappa = 1$, used as an ablation baseline.

4.4 Module 2: Random-Projection Coreset

Storing f for every normal patch yields a large bank. RP-Coreset reduces it with two non-iterative steps.

Random projection. By the Johnson–Lindenstrauss lemma, a random linear projection to dimension d' approximately preserves pairwise Euclidean distances [6]. We draw one random Gaussian matrix $R \in \mathbb{R}^{D\times d'}$ and set $z = R^\top f/\sqrt{d'}$, reducing storage and accelerating nearest-neighbour search while approximately preserving the distances that drive the score.

Uniform subsampling. We keep a fraction r of the projected normal features uniformly at random. The ratio r is a single interpretable parameter: a smaller r gives a smaller, faster bank at some accuracy cost. The resulting bank is $\mathcal{M} = \{m_1, \dots, m_{|\mathcal{M}|}\} \subset \mathbb{R}^{d'}$. Building $\mathcal{M}$ requires one pass over the features, one matrix multiplication, and one random draw, with no iterative selection. The ablation also reports greedy selection [6] to quantify the cost of this simplification.

4.5 Anomaly scoring

Scoring is the standard memory-bank nearest-neighbour rule [5]. For a test patch feature z_i, the patch score is its Euclidean distance to the nearest bank element,

$$s_i = \min_{m\in\mathcal{M}} \| z_i - m \|_2, \tag{2}$$

which is small where the test patch resembles some normal patch and large for anomalies. Arranging s_i over the patch grid and upsampling to the input resolution gives the pixel map S, smoothed by a small Gaussian filter. The image score is the re-weighted maximum [5]. Finally, using a small held-out set of normal images we estimate the mean μ and standard deviation σ of normal image scores and report the normalised score $a = (\hat{a} - \mu)/\sigma$, where $\hat{a}$ is the re-weighted maximum. This rescaling is monotonic, so it does not change AUROC. It only makes the score scale comparable across categories, so one threshold τ at a target false-positive rate on normals transfers without per-category tuning.

4.6 Algorithm and complexity

Algorithm 1 summarises both phases. Let N_p be the number of normal patches, D the fused dimension, d' the projected dimension, and $|\mathcal{M}| = rN_p$.

Algorithm 1 LMF-Core (training-free)

Setup (normal images $\mathcal{X}_n$)
extract ϕ_2, ϕ_3; align; $p\times p$ locally aggregate
compute v_l, σ_c; weights w_l (1); keep top-κ channels
standardise, scale by w_l, concatenate $\Rightarrow$ fused $\{f\}$
draw R; project $z = R^\top f/\sqrt{d'}$ for all patches
$\mathcal{M} \leftarrow$ uniform subsample of $\{z\}$ at ratio r
on held-out normals, estimate μ, σ of image scores
Inference (test image x)
fused feature f; project $z = R^\top f/\sqrt{d'}$
for each patch i **do**
    $s_i \leftarrow \min_{m\in\mathcal{M}} \|z_i - m\|_2$
end for
$S \leftarrow$ upsample/smooth($\{s_i\}$); $\hat{a} \leftarrow$ re-weighted $\max_i s_i$
$a \leftarrow (\hat{a} - \mu)/\sigma$; **return** S, $\mathbb{1}[a > \tau]$

Complexity. Setup is dominated by the projection $O(N_p D d')$; there is *no* $O(|\mathcal{M}|N_p)$ greedy loop. Memory is $O(|\mathcal{M}|\, d') = O(rN_p d')$, controlled directly by r and d'. Per-image inference is one backbone pass plus $O(P\, d')$ for P test patches under approximate nearest-neighbour search [25]. All three parameters (κ, r, d') trade accuracy for memory and latency monotonically.

5. Experimental Protocol

This section specifies the evaluation protocol used for the MVTec AD results in Section 6; it is designed to transfer unchanged to the other benchmarks listed below, whose evaluation we leave to future work.

5.1 Datasets

MVTec AD [2]: 15 categories (10 objects, 5 textures); normal-only training, mixed test sets with pixel masks; the primary benchmark. *VisA* [7]: 12 categories, larger and more varied. *BTAD* [8]: three industrial products with subtle textures. In this paper we report results on MVTec AD only; VisA and BTAD are listed as target benchmarks for future evaluation. We follow the standard protocol: fit on each category's normal training set only, evaluate on its test set, and report the mean over categories.

5.2 Metrics

Image-level AUROC for detection and *pixel-level AUROC* for localisation [12]; both are threshold-free and allow direct comparison with published numbers. For deployment we also consider *efficiency*: memory-bank size, setup time, and per-image latency.

5.3 Implementation and hyper-parameters

Table 2 lists the defaults. Nearest-neighbour search is exact for small banks and approximate (FAISS [25]) for larger ones. Random seeds are fixed; the MVTec AD results in Section 6 are averaged over five random seeds (mean ± std; the per-category and sensitivity studies use a single representative seed). On-device latency and memory benchmarking, needed to fully establish the deployment use case, is left to future work.

Table 2: Default hyper-parameters of LMF-Core.

Symbol	Default	Meaning
backbone	WRN-50 [22]	frozen ImageNet feature extractor
layers	{2,3}	tapped intermediate stages
p	3	local neighbourhood size
κ	0.5	fraction of channels kept (pruning)
d'	128	random-projection dimension
r	0.10	memory-bank subsampling ratio
input	224^2	resize-256 then centre-crop

5.4 Baselines

Published frozen-backbone and trained methods with their reported numbers: SPADE [3], PaDiM [4], CutPaste [11], DRAEM [10], CFLOW-AD [16], FastFlow [15], RD4AD [13], PatchCore [5], SimpleNet [17], Efficien-

tAD [18]. We additionally report internal ablations of LMF-Core (Section 6).

6. Results and Discussion

Note: all LMF-Core numbers are our measured results on MVTec AD (mean over the 15 categories, five-seed mean ± std for the main comparison and ablation (single seed for the per-category and sensitivity studies), $\kappa = 0.5$, $d' = 128$, $r = 0.1$ *unless stated otherwise); baseline numbers are as reported in the cited papers.*

6.1 Main comparison

Table 3: Mean image- and pixel-level AUROC (%) on MVTec AD (15 categories). Baselines as reported by the cited works; LMF-Core is our measured mean.

Method	Training-free	Image AUROC	Pixel AUROC
SPADE [3]	yes	85.5	96.0
PaDiM [4]	yes	95.3	97.5
CutPaste [11]	no	95.2	96.0
DRAEM [10]	no	98.0	97.3
CFLOW-AD [16]	no	98.3	98.6
FastFlow [15]	no	99.4	98.5
RD4AD [13]	no	98.5	97.8
PatchCore [5]	yes	99.1	98.1
SimpleNet [17]	no	99.6	98.1
LMF-Core (ours)	yes	96.3 ± 0.2	96.8 ± 0.1

LMF-Core attains 96.3% image-level and 96.8% pixel-level mean AUROC. Among *training-free* methods it exceeds SPADE and is above PaDiM on detection, marginally below on localisation. It stays below the greedy-coreset PatchCore and the strongest *trained* methods. The gap to PatchCore follows from the design: LMF-Core projects features to $d' = 128$ and prunes half the channels ($\kappa = 0.5$), trading a small accuracy margin for a compact, directly controllable memory bank of about 8.8 MB. The per-category breakdown (Table 4) shows the residual difficulty concentrates on screw (86.4) and pill (87.1), where pose and appearance variation challenge local memory-bank matching, a known limitation of this method family. Increasing d', κ, or r recovers accuracy at the cost of memory.

Table 4: Per-category image- and pixel-level AUROC (%) of LMF-Core (full) on MVTec AD (single seed, $\kappa = 0.5$, $d' = 128$, $r = 0.1$).

Category	Image AUROC	Pixel AUROC
Bottle	100.0	97.4
Cable	98.8	97.6
capsule	92.5	98.2
carpet	99.5	98.4
grid	94.1	95.3
hazelnut	99.4	97.6
leather	100.0	98.6
metal_nut	96.3	97.9
pill	87.1	95.8
screw	86.4	97.2
tile	98.3	93.9
toothbrush	98.1	98.3
transistor	99.8	97.4
wood	98.3	91.8
zipper	97.2	96.5
Mean	96.4	96.8

6.2 Ablation of the two modules

VG-Fusion improves the mean by +0.16 (image) and +0.50 (pixel) AUROC over the equal-weight baseline (five-seed mean ± std: baseline 96.17±0.10 / 96.32±0.04; + VG-Fusion 96.33±0.23 / 96.82±0.07). The localisation (pixel) gain is well outside the seed spread, whereas the image-level change is within it. Per-category (single seed), the image gain concentrates on a few categories (e.g. +5.4 on screw, +2.6 on cable, +1.9 on grid) but is partly offset by a few where the $\kappa = 0.5$ pruning is too aggressive (pill −5.3, metal_nut −2.7), which motivates a per-category κ as future work. Replacing greedy selection with the random-projection coreset changes accuracy by −0.04/0.00 AUROC at a comparable bank size — within the five-seed spread-confirming that the non-iterative RP-Coreset matches greedy selection while removing its iterative optimisation loop.

A note on score calibration. We also tested a local-density-calibrated score that divides the nearest-neighbour distance by the density of the bank around the matched element. On MVTec AD this consistently *reduced* both image- and pixel-level AUROC: true anomalies often match bank elements at the sparse boundary of the normal manifold, where the large local density suppresses their score. We therefore report it only as a negative result and keep the plain nearest-neighbour distance of Eq. (2) in the final method.

Table 5: Ablation on MVTec AD (mean over 15 categories). Each row adds one module to the equal-weight greedy-coreset baseline; greedy and random use a comparable bank size.

Variant	Img. AUROC	Pix. AUROC	$\lvert\mathcal{M}\rvert$ (k)	Bank (MB)
Baseline (equal feat. + greedy)	96.17 ± 0.10	96.32 ± 0.04	16.4	8.4
+ VG-Fusion (M1, greedy)	96.33 ± 0.23	96.82 ± 0.07	16.4	8.4
+ RP-Coreset (M1+M2, full)	96.29 ± 0.25	96.82 ± 0.07	17.1	8.8

6.3. Sensitivity

Table 6 and Fig. 2 report the sensitivity to the three parameters (κ, d', r). Detection (image AUROC) is the sensitive metric; localisation stays within 96.2–96.9% pixel AUROC across the entire sweep. Channel-keep κ is highest at the intermediate value 0.5 (96.3): pruning too aggressively ($\kappa = 0.25$, 96.0) discards useful channels, whereas keeping all of them ($\kappa = 1.0$, 95.8) re-admits the low-variance channels that VG-Fusion is designed to suppress. Raising the projection dimension helps monotonically but with diminishing returns ($d' = 64 \rightarrow 96.1$, $128 \rightarrow 96.3$, $256 \rightarrow 96.6$). Figure 3 plots AUROC against the subsampling ratio r: it rises steeply then saturates, from 91.7 at $r = 0.01$ to 96.3 at $r = 0.1$, after which doubling the bank again ($r = 0.25 \rightarrow 0.5$) adds only 0.2 point ($97.4 \rightarrow 97.5$). A small r therefore meets the accuracy target while keeping the bank small. These results provide practical justification for RP-Coreset: a single interpretable parameter instead of an iterative greedy procedure. The bank itself is small: at the default setting it occupies only about 8.8 MB (Table 5), and the random-projection construction avoids the iterative greedy loop en-

tirely. These sensitivity values are five-seed means with a per-point standard deviation of 0.1–0.4 image AUROC (Table 6); image differences below about 0.3 point (e.g. κ = 0.25 vs 0.5, adjacent d′ steps) are therefore within seed noise and should be read as indicative. The plotted markers in Figs 2–3 use a single representative seed.

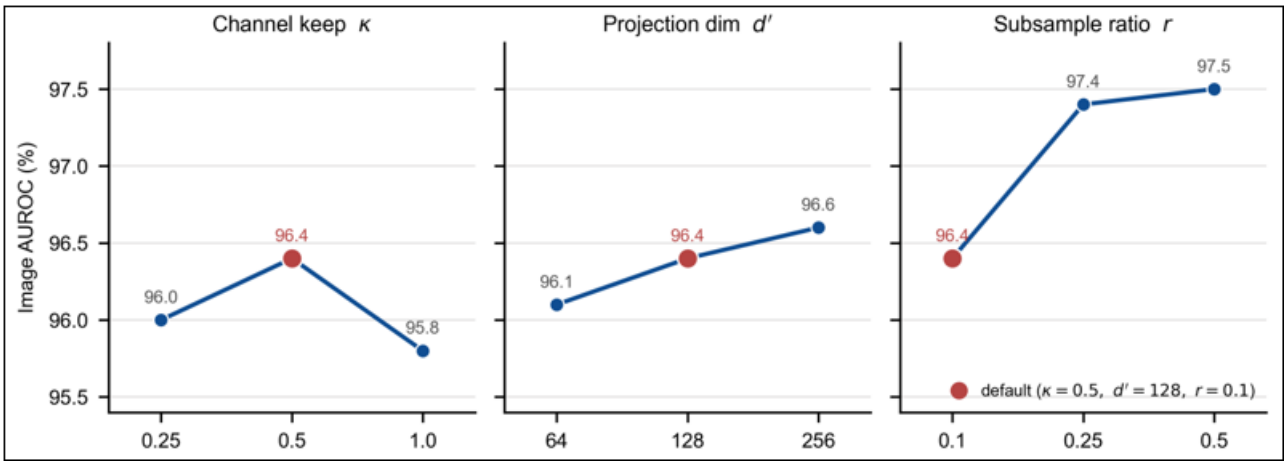

Figure 2: One-factor-at-a-time sensitivity of image-level AUROC to the three parameters on MVTec AD: channel-keep κ, projection dimension d', and subsampling ratio r. The red marker is the default operating point ($\kappa = 0.5,\ d' = 128,\ r = 0.1$), shared by all three panels. κ is highest at the intermediate 0.5, whereas d' and r mainly trade memory for accuracy

Table 6: Sensitivity to channel-keep κ, projection dim d', and ratio r (image AUROC %, MVTec AD five-seed mean ± std). The $\kappa = 0.5, d' = 128, r = 0.1$ row matches the main result (96.3).

κ	d'	r	Img. AUROC
0.25	128	0.10	96.0 ± 0.23
0.50	128	0.10	96.3 ± 0.25
1.00	128	0.10	95.8 ± 0.18
0.50	64	0.10	96.1 ± 0.22
0.50	256	0.10	96.6 ± 0.20
0.50	128	0.25	97.4 ± 0.11

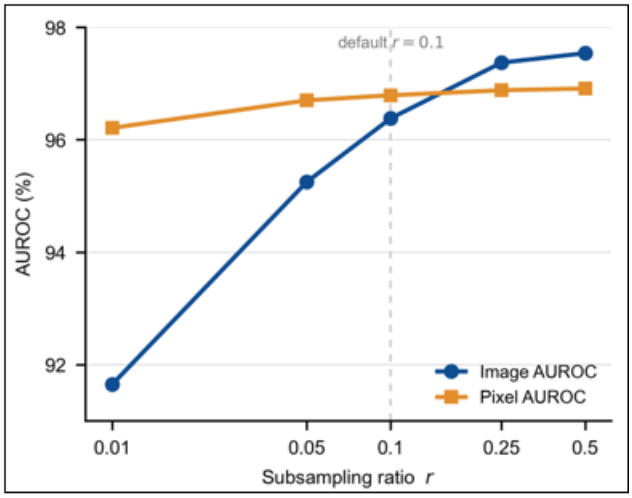

Figure 3: Image- and pixel-level AUROC vs. subsampling ratio r on MVTec AD ($\kappa = 0.5$, $d' = 128$). Image-level accuracy rises steeply then saturates, so a small r keeps the memory bank small at little accuracy cost; pixel-level accuracy is nearly flat

6.4 Qualitative localisation

Figure 4 shows a qualitative result: an input surface with a real defect, the ground-truth mask, and the predicted anomaly heatmap overlaid on the input. The warm high-response region coincides with the defect, confirming that the patch scores of Eq. (2) localise the anomaly directly, with no segmentation training.

Figure 4: Qualitative anomaly localisation on an industrial surface sample. Left: input image with a real surface defect. Middle: ground-truth mask. Right: predicted anomaly heatmap overlaid on the input; the warm region coincides with the defect. The map is produced directly from the nearest-neighbour patch scores, without segmentation training.

6.5 Discussion

The framework trades a small accuracy margin for a large reduction in setup cost and a directly controllable footprint, both important for embedded inspection [18]. VG-Fusion is a statistics-only change that adapts the representation per category; the per-category ablation gains vary across categories (Section 6.2), consistent with category-dependent layer and channel informativeness, although we do not directly visualise the learned weights here. Together with the score-calibration negative result above, this supports a minimal pipeline: the two modules that improve accuracy act on the *features* and the *bank*, not on post-processing of the distance.

7. Limitations and Future Work

LMF-Core inherits the limits of frozen-backbone local memory-bank methods. *(1) Logical anomalies:* local per-patch scoring detects texture and structural defects but is less effective on global-constraint violations (e.g., wrong count) [1]; a complementary global descriptor is needed. *(2) Backbone domain gap:* ImageNet features may be poor for modalities far from natural images (X-ray, infrared), needing a domain-adapted or self-supervised backbone.

(3) 3D/multimodal defects: pure RGB features miss geometry visible only in depth or point clouds [20]; extending the fused bank to depth is a natural step. *(4) Sampling variance:* uniform subsampling adds run-to-run variance (±0.25 image AUROC over five seeds); stratified or density-aware sampling could reduce it without a greedy loop. Future work: a global descriptor for logical anomalies; extension to few-/zero-shot via CLIP-based backbones [24], [19]; and an RGB+depth multimodal bank [20].

8. Conclusion

We presented LMF-Core, a training-free framework for unsupervised industrial visual inspection. It retains the core properties of frozen-backbone memory-bank methods and adds two coordinated, statistics-only modules: variance-guided fusion for more discriminative features, and a random-projection coreset for a low-cost, directly controllable memory bank. A negative result is also reported: calibrating the score by local density did not improve accuracy on MVTec AD. We provide the full algorithm, a complexity analysis, a reproducible protocol with results on MVTec AD, an ablation isolating each module, and qualitative localisation. The design targets resource-constrained inspection hardware, with concrete extensions toward logical, few-shot, and 3D anomaly detection.

References

[1] J. Liu, G. Xie, J. Wang, S. Li, C. Wang, F. Zheng, and Y. Jin, "Deep Industrial Image Anomaly Detection: A Survey," *Machine Intelligence Research*, vol. 21, pp. 104–135, 2024.

[2] P. Bergmann, M. Fauser, D. Sattlegger, and C. Steger, "MVTec AD — A Comprehensive Real-World Dataset for Unsupervised Anomaly Detection," in *Proc. IEEE/CVF CVPR*, 2019, pp. 9592–9600.

[3] N. Cohen and Y. Hoshen, "Sub-Image Anomaly Detection with Deep Pyramid Correspondences (SPADE)," *arXiv:2005.02357*, 2020.

[4] T. Defard, A. Setkov, A. Loesch, and R. Audigier, "PaDiM: A Patch Distribution Modeling Framework for Anomaly Detection and Localization," in *Proc. ICPR Workshops*, 2021.

[5] K. Roth, L. Pemula, J. Zepeda, B. Schölkopf, T. Brox, and P. Gehler, "Towards Total Recall in Industrial Anomaly Detection (PatchCore)," in *Proc. IEEE/CVF CVPR*, 2022, pp. 14318–14328.

[6] O. Sener and S. Savarese, "Active Learning for Convolutional Neural Networks: A Core-Set Approach," in *Proc. ICLR*, 2018.

[7] Y. Zou, J. Jeong, L. Pemula, D. Zhang, and O. Dabeer, "SPot-the-Difference Self-Supervised Pre-training for Anomaly Detection and Segmentation (VisA)," in *Proc. ECCV*, 2022.

[8] P. Mishra, R. Verk, D. Fornasier, C. Piciarelli, and G. L. Foresti, "VT-ADL: A Vision Transformer Network for Image Anomaly Detection and Localization (BTAD)," in *Proc. IEEE Int. Symp. Industrial Electronics (ISIE)*, 2021.

[9] S. Akcay, A. Atapour-Abarghouei, and T. P. Breckon, "GANomaly: Semi-Supervised Anomaly Detection via Adversarial Training," in *Proc. ACCV*, 2018.

[10] V. Zavrtanik, M. Kristan, and D. Skočaj, "DRAEM — A Discriminatively Trained Reconstruction Embedding for Surface Anomaly Detection," in *Proc. IEEE/CVF ICCV*, 2021.

[11] C.-L. Li, K. Sohn, J. Yoon, and T. Pfister, "CutPaste: Self-Supervised Learning for Anomaly Detection and Localization," in *Proc. IEEE/CVF CVPR*, 2021.

[12] P. Bergmann, M. Fauser, D. Sattlegger, and C. Steger, "Uninformed Students: Student–Teacher Anomaly Detection with Discriminative Latent Embeddings," in *Proc. IEEE/CVF CVPR*, 2020.

[13] H. Deng and X. Li, "Anomaly Detection via Reverse Distillation from One-Class Embedding," in *Proc. IEEE/CVF CVPR*, 2022.

[14] M. Rudolph, B. Wandt, and B. Rosenhahn, "Same Same But DifferNet: Semi-Supervised Defect Detection with Normalizing Flows," in *Proc. IEEE/CVF WACV*, 2021.

[15] J. Yu, Y. Zheng, X. Wang, W. Li, Y. Wu, R. Zhao, and L. Wu, "FastFlow: Unsupervised Anomaly Detection and Localization via 2D Normalizing Flows," *arXiv:2111.07677*, 2021.

[16] D. Gudovskiy, S. Ishizaka, and K. Kozuka, "CFLOW-AD: Real-Time Unsupervised Anomaly Detection with Localization via Conditional Normalizing Flows," in *Proc. IEEE/CVF WACV*, 2022.

[17] Z. Liu, Y. Zhou, Y. Xu, and Z. Wang, "SimpleNet: A Simple Network for Image Anomaly Detection and Localization," in *Proc. IEEE/CVF CVPR*, 2023.

[18] K. Batzner, L. Heckler, and R. König, "EfficientAD: Accurate Visual Anomaly Detection at Millisecond-Level Latencies," in *Proc. IEEE/CVF WACV*, 2024.

[19] J. Jeong, Y. Zou, T. Kim, D. Zhang, A. Ravichandran, and O. Dabeer, "WinCLIP: Zero-/Few-Shot Anomaly Classification and Segmentation," in *Proc. IEEE/CVF CVPR*, 2023.

[20] P. Bergmann, X. Jin, D. Sattlegger, and C. Steger, "The MVTec 3D-AD Dataset for Unsupervised 3D Anomaly Detection and Localization," in *Proc. VISAPP*, 2022.

[21] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A Large-Scale Hierarchical Image Database," in *Proc. IEEE CVPR*, 2009.

[22] S. Zagoruyko and N. Komodakis, "Wide Residual Networks," in *Proc. BMVC*, 2016.

[23] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *Proc. IEEE CVPR*, 2016, pp. 770–778.

[24] A. Radford *et al.*, "Learning Transferable Visual Models From Natural Language Supervision (CLIP)," in *Proc. ICML*, 2021.

[25] J. Johnson, M. Douze, and H. Jégou, "Billion-Scale Similarity Search with GPUs (FAISS)," *IEEE Trans. Big Data*, vol. 7, no. 3, pp. 535–547, 2021.

[26] X. Tao, X. Gong, X. Zhang, S. Yan, and C. Adak, "Deep Learning for Unsupervised Anomaly Localization in Industrial Images: A Survey," *IEEE Trans. Instrumentation and Measurement*, vol. 71, pp. 1–21, 2022.