International Journal of Emerging Technologies and Advanced Applications

Few-Normal-Shot Tile Anomaly Detection with Lightweight Multiscale Memories

Ziwen Co., Limited, Beijing,China[1]
buddhalwc@126.com

***Abstract*—Surface inspection of ceramic tiles is constrained by scarce representative defects and costly pixel annotation. This study evaluates lightweight memory banks fitted with only a few normal images. A frozen pretrained ResNet18 provides multiscale local descriptors, which are optionally rescaled using normal-image channel statistics and layer widths, projected into a compact space, and stored in a fixed-capacity coreset. The initial evaluation comprises 144 configurations on the Tile, Grid, and Wood categories of MVTec AD, with one, four, or sixteen normal images and three support-set seeds. For four-shot Tile, the normalized multiscale model achieves an image-level area under the receiver operating characteristic curve of 97.92 percent and a pixel-level value of 93.61 percent. Relative to a same-backbone adaptation of the authors' PatchCore code, these values improve by 8.25 and 0.94 percentage points, whereas pixel average precision decreases by 2.67 points. A larger WideResNet50 baseline remains more accurate. Supplementary controlled diagnostics do not establish that the combined normalization components outperform simpler variants. The contribution is a reproducible empirical assessment of accuracy and computational tradeoffs, rather than a claim of uniformly superior localization.**

***Index Terms*—Ceramic tile inspection, Anomaly detection, Few-shot learning, Multiscale features, Memory bank**

I. Introduction

Ceramic tile inspection requires both deciding whether an image is abnormal and locating the affected region. For new textures or small production batches, verified normal images can be easier to obtain than representative defects. Defect appearance, size, and position may change, making a fixed set of labeled damage classes expensive to maintain. Normal-only anomaly detection offers a complementary formulation: a model describes normal appearance and assigns high scores to deviations without learning a separate detector for each defect type.

This study considers a few-normal-shot setting. For each target category, exactly k normal images fit all target-domain statistics and memory contents. Unselected normal training images are not accessed by model fitting. This differs from fully unsupervised representation learning: the feature extractor uses supervised ImageNet pretraining. The system produces an image anomaly score and a dense anomaly map, not crack/pit class labels or bounding-box detections. Bounding-box mean average precision is therefore not an appropriate evaluation metric.

MVTec AD provides official normal training images, separate test images, and pixel-level test masks [1]. Its Tile category permits reproducible tile-surface experiments without unverified annotations. Grid and Wood provide additional texture categories, but each is fitted independently. Their results assess applicability across categories, not train-on-one-domain, test-on-another generalization.

The research question is constrained: under a small backbone, a low-dimensional descriptor, and a fixed memory budget, how do additional feature levels and normal-channel scaling affect anomaly detection? Multiscale features, random projection, and coreset selection are established components. The contribution is their controlled evaluation with unfavorable outcomes and implementation boundaries. Specifically, this work defines paired normal-support experiments, examines image-level and pixel-level tradeoffs, and adds diagnostic ablations and uncertainty estimates that test the necessity of the combined scaling components. It does not claim a new general-purpose state of the art.

II. Related Work

Distribution-based and memory-based methods model normal appearance using pretrained features. PaDiM estimates a multivariate Gaussian distribution at each spatial position [2], while PatchCore retains representative normal patch descriptors for nearest-neighbor matching [3]. Extremely small support sets make position-specific covariance estimation difficult. A shared patch memory reduces dependence on exact spatial alignment, although its quality depends on representation, distance definition, and compression.

Other approaches learn through synthetic anomalies, reconstruction, or distillation. CutPaste creates local image perturbations [4], and DRAEM combines reconstruction with discriminative localization [5]. Student–teacher feature pyramid matching compares multiple representation levels [6]. DifferNet and CFLOW model normal features with normalizing flows [7], [8]. Reverse distillation, SimpleNet, DeSTSeg, and revised reverse distillation further explore feature recovery

ISSN:3006-2985(Print)
ISSN:3006-9297(Online)

Received: Jul. 27, 2026
Revised: Aug. 20, 2026
Accepted: Aug. 21, 2026

and discriminative learning [9]–[12]. Their optimization and auxiliary-data requirements differ from direct memory fitting; fitting times cannot be compared without matching those conditions.

Foundation models extend zero-shot and few-normal-shot inspection. WinCLIP combines local image representations with language [13]; PromptAD learns prompts from normal examples [14]; AnomalyCLIP learns object-agnostic abnormality semantics [15]. AnomalyDINO, UniVAD, and differentiable feature matching demonstrate the importance of pretrained representations and matching design [16]–[18]. They motivate a stronger-backbone comparator here but are not reproduced in the present experiments.

Efficiency is addressed by EfficientAD [19], while RealNet investigates feature selection and realistic synthetic anomalies [20]. Few-shot PatchCore analysis finds value in hyperparameter adaptation, while image augmentation does not guarantee improvement [21]. ObjectCore extends inspection to logical anomalies involving object arrangements [22]. The present study concerns surface appearance in textures, not logical consistency or object composition.

III. Lightweight Multiscale Memory

A. Task and Feature Extraction

Let $\mathcal{D} = \{x_i\}_{i=1}^{k}$ denote a normal support set and let x be a test image. Images are bilinearly resized to 256×256 and normalized with ImageNet channel means and standard deviations. No border crop is applied. The ResNet18 backbone is frozen in evaluation mode.

Features are extracted from layers 1, 2, and 3, with 64, 128, and 256 channels. Each map undergoes local 3×3 average pooling and bilinear alignment to a 32×32 grid. The descriptor at location p is

$$f(p) = \mathrm{concat}_{l=1}^{3}[A_l(F_l)(p)], \tag{1}$$

where F_l is the feature map and A_l denotes aggregation and alignment. Each image supplies 1,024 descriptors of dimension 448. Multiscale refers to backbone feature levels, not multiple input resolutions or test-time augmentation.

Shallow features retain local texture information; deeper features have larger receptive fields. Combining them may help localization, but different widths and numerical scales can distort a Euclidean distance. These considerations motivate controlled variants rather than assuming every component helps. Fig. 1 summarizes fitting and inference.

B. Normal-Channel Scaling and Layer Balancing

For each channel, its mean $\mu_{l,c}$ and standard deviation $\sigma_{l,c}$ are computed over local descriptors from the selected k images. Let q_l be the tenth percentile of channel standard deviations in layer l, with C_l channels. The transform is

$$z_{l,c}(p) = \frac{f_{l,c}(p) - \mu_{l,c}}{s_{l,c}}, \tag{2}$$
$$s_{l,c} = \max(\sigma_{l,c}, q_l, 10^{-3})\sqrt{C_l}.$$

The floor limits amplification of almost constant channels. Division by $\sqrt{C_l}$ approximately balances layers of different widths. These are fixed design choices, not parameters selected using test labels.

No unselected training image or test image updates these statistics. A common mean shift cancels in Euclidean distances after the same linear mapping: channel scales and layer weights change the distance geometry, not centering itself. Small normal variance does not prove defect relevance. Scaling may amplify nuisance variation, making an unnormalized multiscale comparator essential.

C. Projection and Memory Construction

A fixed Gaussian projection reduces the descriptor dimension:

$$u(p) = z(p)R, \qquad R \in \mathbb{R}^{448\times 128}, \tag{3}$$

with independent entries distributed as $\mathcal{N}(0, 1/128)$ and seed 20260921. The projection is not optimized on anomalies.

The $1024k$ normal descriptors are further projected to a 32-dimensional selection space with seed 179. Greedy farthest-point sampling selects 512 representatives. The first point is farthest from the sample mean; each subsequent point maximizes its distance to the closest selected point. The saved memory contains the corresponding 128-dimensional descriptors, not the selection-space vectors. The local score is

$$d(p) = \min_{m\in\mathcal{M}} \|u(p) - m\|_2, \qquad |\mathcal{M}| = 512. \tag{4}$$

Patches from one normal image are spatially dependent memory elements, not additional independent support images. A one-shot model still has only one normal support image.

D. Anomaly Scores

The image score is the maximum raw patch distance before smoothing. For localization, the 32×32 map is bilinearly upsampled to 256×256 and Gaussian-smoothed with standard deviation four pixels. Image scores and pixel maps are evaluated separately. No binary threshold is selected from test labels, and no deployment-level decision is claimed without calibration.

IV. Experimental Protocol

A. Data and Integrity Checks

Official MVTec AD splits are used [1]. Tile is the primary category; Grid and Wood are additional independently fitted texture experiments. Their normal training pools contain 230, 264, and 247 images, and test sets contain 117, 78, and 79 images. Tile has 33 normal and 84 anomalous test images, Grid has 21 and 57, and Wood has 19 and 60. Fig. 2 shows the split counts; only k normal images are fitted per run.

Tile defects comprise crack, glue strip, gray stroke, oil, and rough surface. These labels do not correspond directly to arbitrary building-site damage classes. The study does not relabel them into a four-class detector. Official masks are resized by nearest-neighbor interpolation: localization metrics concern the common grid rather than original-resolution boundaries.

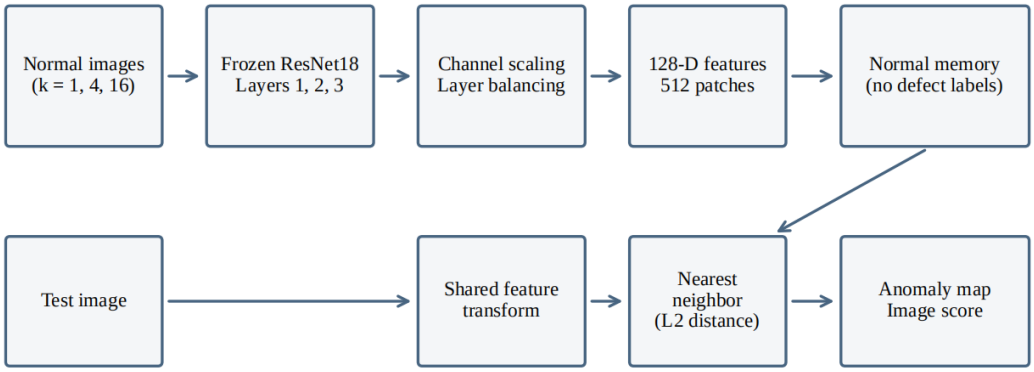

Fig. 1: Normal-only fitting and inference of the normalized multiscale memory. Target-domain statistics use only the selected normal support images.

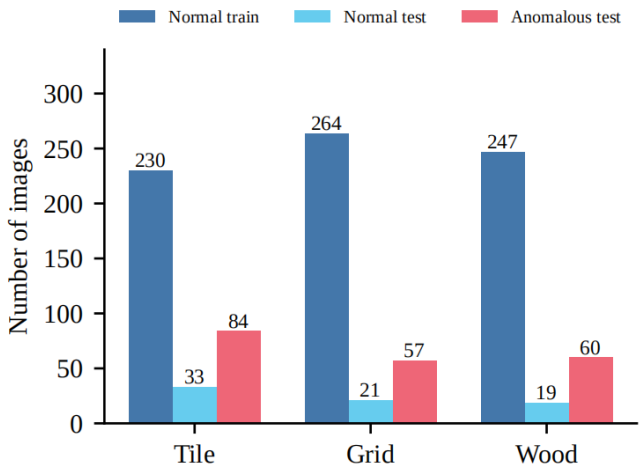

Fig. 2: Official split sizes. Each fitted model uses only its selected k normal training images.

A public mirror accelerated initial transfer and was checked against an independently downloaded official archive. All 1,222 selected image and mask files matched by SHA256. No byte-level training–test duplication was found, and anomalous test masks were complete. These checks establish file consistency, not the absence of all scene correlations. Full hashes and sources accompany the reproducibility materials. The data are used under CC BY-NC-SA 4.0. No additional building-site images enter fitting or evaluation.

B. Paired Support Sampling

Seeds 11, 22, and 33 shuffle each category's normal training pool. The first 1, 4, or 16 images form nested support sets. All methods share identical support images for a category, seed, and size, while fitted statistics and memories remain independent. Evaluating the same test image under different supports does not create extra independent test images.

The primary endpoint was fixed locally before real-data evaluation: mean pixel AUROC on four-shot Tile over the three seeds. Image size, memory capacity, projection dimension, smoothing, and seeds were frozen then. This local freeze is not public preregistration.

The initial controlled design has 3 categories $\times 3$ support sizes $\times 3$ seeds $\times 4$ methods, giving 108 configurations. A same-backbone author-code comparator adds 27. Nine larger-backbone Tile configurations were added after preliminary results were inspected, without changing the proposed pipeline. All 144 initial results are retained. Supplementary diagnostics add nine module fits and nine score-only recomputations, explicitly treated as post-hoc evidence.

C. Comparators and Adaptations

PatchMemory (PM) uses layers 2 and 3. MultiScaleMemory (MSM) adds layer 1. NormalizedMultiScaleMemory (NMSM) also applies (2). All three use 128-dimensional descriptors and 512 memory points. The feature cache saves repeated backbone computation without expanding any support set.

PaDiM-R is a regularized adaptation with the same ResNet18 features. It selects 64 channels using a fixed seed, estimates a covariance per spatial position, and adds $0.01I$. With one image, sample covariance is zero before regularization. This is an extreme-few-shot adaptation, not the original paper's standard setting.

PatchCore-A reuses aggregation, approximate greedy coreset sampling, and map postprocessing from the authors' code, with ResNet18, 128 output dimensions, and 512 points. PatchCore-W uses WideResNet50-2, 1,024 output dimensions, and the same point count. Both retain common image size and supports. The source commit is `fcaa92f...cbcad` (full hash in the supplement).

Exact CUDA distances replace FAISS search. These configurations use squared L2 patch scores and maximum raw-grid image scores without the paper's neighborhood reweighting. They are author-code adaptations, not full reproductions of

TABLE I: Four-shot Tile results: mean and sample standard deviation over three support seeds.

Configuration	Image AUROC (%)	Pixel AUROC (%)	Pixel AP (%)
PaDiM-R	89.08 ± 2.18	89.17 ± 1.17	36.06 ± 1.91
PM	96.26 ± 1.50	92.45 ± 0.88	40.21 ± 1.59
PatchCore-A	89.67 ± 2.05	92.67 ± 0.66	42.03 ± 1.69
MSM	97.13 ± 0.76	93.86 ± 0.82	46.02 ± 2.25
NMSM	97.92 ± 1.21	93.61 ± 0.09	39.36 ± 1.74
PatchCore-W	99.33 ± 0.74	96.11 ± 0.20	56.98 ± 0.99

every original setting. Equal point counts do not imply equal byte budgets for different descriptor dimensions.

D. Metrics and Computing Environment

Image AUROC, pooled pixel AUROC, and pooled pixel average precision (AP) are reported as means and sample standard deviations over three support seeds. Pixel AP complements AUROC under substantial background imbalance. Pixel scores are pooled across all test images, rather than averaging per-image AUROCs. Regional AUPRO and calibrated operating-point results are not reported.

Experiments use an NVIDIA GeForce RTX 5070 Laptop GPU with approximately 12 GB memory and PyTorch 2.10.0 with CUDA 12.8. TF32 is disabled. Feature extraction, memory selection, and nearest-neighbor search run on the GPU; metric sorting runs on the CPU. PaDiM covariance inversion uses the GPU, while quadratic-form scoring uses NumPy. Fitting, cached inference, and independent single-image latency are recorded separately.

V. Results and Analysis

A. Primary Tile Experiment

Table I reports the predefined four-shot Tile setting. NMSM reaches $97.92 \pm 1.21\%$ image AUROC and $93.61 \pm 0.09\%$ pixel AUROC. Absolute gains over PM are 1.66 and 1.16 percentage points; over PatchCore-A, 8.25 and 0.94 points. These are percentage-point differences, not relative percentage gains. Low variation over three supports cannot establish deployment stability.

The gains are not uniform. NMSM pixel AP is $39.36 \pm 1.74\%$, below PatchCore-A at $42.03 \pm 1.69\%$ and MSM at $46.02 \pm 2.25\%$. Better overall ranking can coexist with poorer precision among high-scoring pixels. A small AUROC increase does not imply better segmentation at every threshold.

PatchCore-W attains $99.33 \pm 0.74\%$ image AUROC, $96.11 \pm 0.20\%$ pixel AUROC, and $56.98 \pm 0.99\%$ pixel AP, all above NMSM. The difference reflects backbone capacity, aggregation, and descriptor dimension jointly. The lightweight model is a budget-dependent configuration, not an accuracy replacement for the larger comparator.

B. Number of Normal Images

Fig. 3 reports all support sizes. From one to sixteen images, NMSM Tile image AUROC rises from 97.11 to 98.68 percent and pixel AUROC from 92.71 to 94.15 percent. MSM pixel AP rises from 41.38 to 46.94 percent. Additional supports can broaden normal-appearance coverage, but the fixed 512-point memory must still compress it.

One normal image may contain representative texture or omit important variation. Three seeds do not cover all supports. Error bars describe support-seed standard deviations, not test-image confidence intervals. The primary four-shot setting is not replaced by the most favorable curve point.

C. Multiscale and Scaling Effects

The original PM-to-MSM comparison raises four-shot Tile pixel AUROC from 92.45 to 93.86 percent and AP from 40.21 to 46.02 percent. It also changes input dimension and projection-row correspondence. This is a pipeline-level comparison, not an isolated shallow-layer test.

Supplementary PM-shared uses MSM's 448×128 projection and zeros layer-1 coordinates in both normal and test descriptors. Pixel AUROC is 92.73 percent and AP 41.97 percent. The remaining difference supports shallow information in this configuration, not across every projection seed. Its memory is reselected, so the result includes changed coreset contents.

MSM-to-NMSM retains feature levels and the projection matrix. Rescaling and the resulting coreset increase image AUROC from 97.13 to 97.92 percent but reduce pixel AUROC by 0.25 points and AP by 6.66 points. Balancing layer energy and improving precise localization are not interchangeable objectives.

D. Applicability Across Texture Categories

Fig. 4 reports independently fitted categories. Four-shot Grid is a clear weakness: NMSM pixel AUROC is 67.88 percent, versus 63.42 for PM and 63.02 for PatchCore-A, while NMSM pixel AP is only about 1.73 percent. Relative improvement does not remove failure. Repetitive texture, resizing, and incomplete coverage are plausible explanations whose causal roles have not been isolated.

For four-shot Wood, NMSM reaches 98.83 percent image AUROC, 92.20 percent pixel AUROC, and 36.02 percent AP. MSM reaches 92.96 percent pixel AUROC and 44.37 percent AP. The localization tradeoff is not restricted to Tile. These three categories do not constitute the full fifteen-category benchmark.

E. Uncertainty of the Primary Endpoint

A supplementary paired image-cluster bootstrap targets pooled pixel AUROC. Each of 1,000 replicates resamples normal and anomalous test images within their strata, retaining all pixels of each sampled image. The same image indices are used across methods and the three fixed support seeds. Spatially correlated pixels are not treated as independent observations.

Computation uses 8,192 quantile bins per fitted model, retaining positive and negative pixel counts per image and bin. Half the within-bin positive–negative pair count divided by the total positive–negative pair count bounds the AUROC approximation error. Both comparator bounds are propagated outward into the percentile interval. On the original sample,

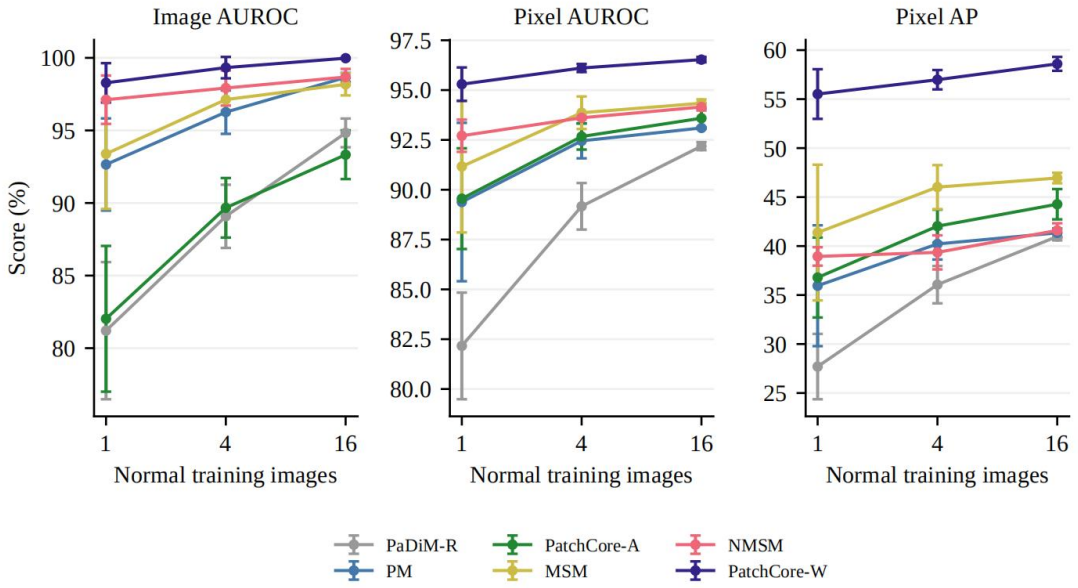

Fig. 3: Tile performance for all predefined normal support sizes. Error bars indicate sample standard deviations over three support seeds.

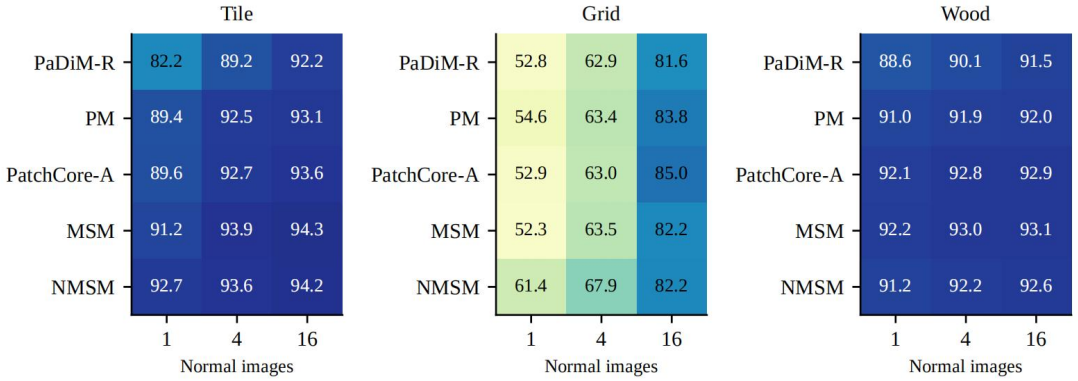

Fig. 4: Pixel AUROC in independently fitted texture categories. Means are over three support seeds; no cross-category transfer is performed.

each model's bound is below 0.0045 percentage points. Table point estimates retain exact sorting.

Conditional 95 percent intervals for NMSM minus its comparators, in percentage points, are $[0.40, 2.05]$ against PM, $[0.20, 1.78]$ against PatchCore-A, $[-0.70, 0.26]$ against MSM, and $[-3.23, -1.76]$ against PatchCore-W (Fig. 5). They do not support superiority over MSM or the larger backbone. They condition on three supports and do not cover new supports, domains, or method selection. Multiple comparisons are not adjusted; intervals are descriptive.

F. Fixed Qualitative Examples

Fig. 6 uses four-shot seed 11 and the lexicographically first test image of each Tile defect. Examples are not chosen by performance. Each method uses a display scale fixed across its Tile test maps at the 99.5th score percentile. Colors are spatial responses, not calibrated probabilities or directly comparable confidence.

Examples should be read alongside AP. Responses outside boundaries, weak responses inside defects, and edge effects may look plausible while producing imprecise segmentation. Fixed Gaussian smoothing reduces grid artifacts but spreads responses. No parameter is changed to improve an individual displayed image.

G. Supplementary Module Diagnostics

Table II reports post-hoc diagnostics on the original three four-shot Tile supports. Scale-only applies channel-standard-deviation scaling without layer balancing. Balance-only divides by the square root of layer width without channel scaling. Both use the shared projection and reselect 512 points, so comparisons include the induced coreset changes.

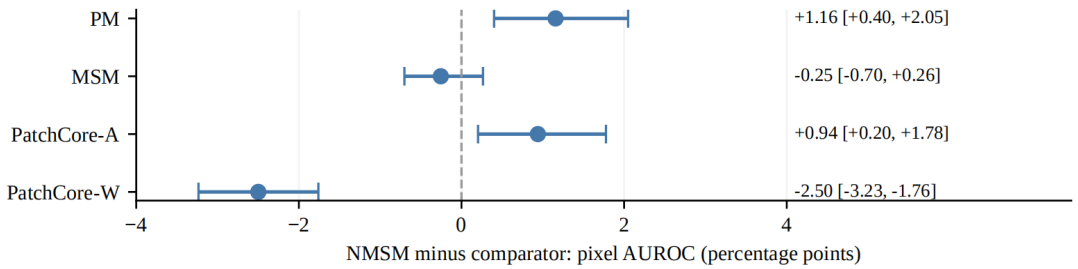

Fig. 5: Conditional 95 percent intervals for primary pixel-AUROC differences. Image-cluster resampling retains spatial dependence and numerical binning bounds are propagated conservatively.

TABLE II: Post-hoc four-shot Tile diagnostics: mean and sample standard deviation over the same three supports.

Configuration	Image AUROC (%)	Pixel AUROC (%)	Pixel AP (%)
PM-shared	96.01 ± 2.05	92.73 ± 1.32	41.97 ± 2.91
Scale-only	98.16 ± 0.81	93.22 ± 0.13	37.32 ± 1.54
Balance-only	95.65 ± 0.90	93.95 ± 0.74	49.21 ± 2.13
PatchCore-A (L2)	89.67 ± 2.05	92.73 ± 0.66	42.37 ± 1.71
PatchCore-W (L2)	99.33 ± 0.74	96.13 ± 0.20	57.41 ± 0.99
NMSM (squared L2)	97.92 ± 1.21	93.53 ± 0.09	38.89 ± 1.76

Balance-only reaches 93.95 percent pixel AUROC and 49.21 percent AP, both above NMSM. Scale-only reaches 98.16 percent image AUROC, also above the combined model. These findings weaken an argument that both components must be combined. They are not used to replace the primary model after seeing the same test set. Subsequent selection requires independent development and validation.

H. Distance Definition and Smoothing

Initial PatchCore adaptations use squared L2 and the memory variants use L2. Squaring is monotonic and preserves the ranking of maximum raw patch scores. However, squaring before interpolation and smoothing is not equivalent to squaring afterward, so pixel rankings can differ.

With original memories fixed, both PatchCore variants are rescored with L2 and NMSM with squared L2. L2 PatchCore-A reaches 92.73 percent pixel AUROC and 42.37 percent AP; PatchCore-W reaches 96.13 and 57.41 percent. Squared-L2 NMSM reaches 93.53 and 38.89 percent. The principal tradeoffs do not reverse, but the numerical effect confirms that Table I compares full implementations, not isolated feature modules. All nine score-only recomputations are retained.

VI. Computational Cost and Limitations

A. Fitting State and Single-Image Latency

Fitting estimates statistics and selects representatives without backbone gradient updates. No artificial training-loss curve is used as evidence. Fig. 7 reports fitting time excluding backbone extraction and compressed fitted-state size. Disk bytes are not GPU memory and exclude pretrained weights, feature caches, and test outputs. Independent single-image timing avoids equating cached nearest-neighbor time with deployment latency. The Tile four-shot seed-11 model receives a normalized 256×256 image on the GPU and returns a smoothed CPU anomaly map. Timing includes extraction, nearest-neighbor search, upsampling, output transfer, and smoothing. It excludes disk access, decoding, input preprocessing, initialization, and fitting. Each method is warmed up ten times and measured for 100 consecutive executions with CUDA synchronization. Main experiments have finished before this measurement.

Initially, NMSM has median latency 2.59 ms and 95th percentile 2.64 ms; PatchCore-W has 16.12 and 20.31 ms. The median reduction is about 84.0 percent for these implementations, not a hardware-independent speedup or complete camera-to-decision latency.

Supplementary same-backbone timing gives medians of 2.49 ms for MSM, 2.48 ms for NMSM, 3.90 ms for PatchCore-A, and 16.14 ms for PatchCore-W. The MSM–NMSM difference does not support a normalization-based acceleration claim. Author-code comparisons also include aggregation and implementation overhead. Consecutive measurements do not quantify between-session power or temperature variation.

Full ResNet18 and WideResNet50-2 contain 11.69 and 68.88 million parameters. Their executed prefixes through the third feature stage contain 2.78 and 24.86 million, an 88.8 percent reduction. NMSM also uses 128-dimensional instead of 1,024-dimensional memory entries. The speed advantage cannot be attributed to channel scaling alone.

B. Scope and Remaining Evidence Gaps

Only three support seeds are evaluated. Correlated imaging conditions and a single public tile category limit external validity. Grid failures show that the representation is not reliable for every texture. Resizing standardizes computation but may remove fine boundaries; original-resolution or tiled inference requires a separate budget-matched study.

The method depends on ImageNet pretraining and verified normal supports. Contamination could place defects into the

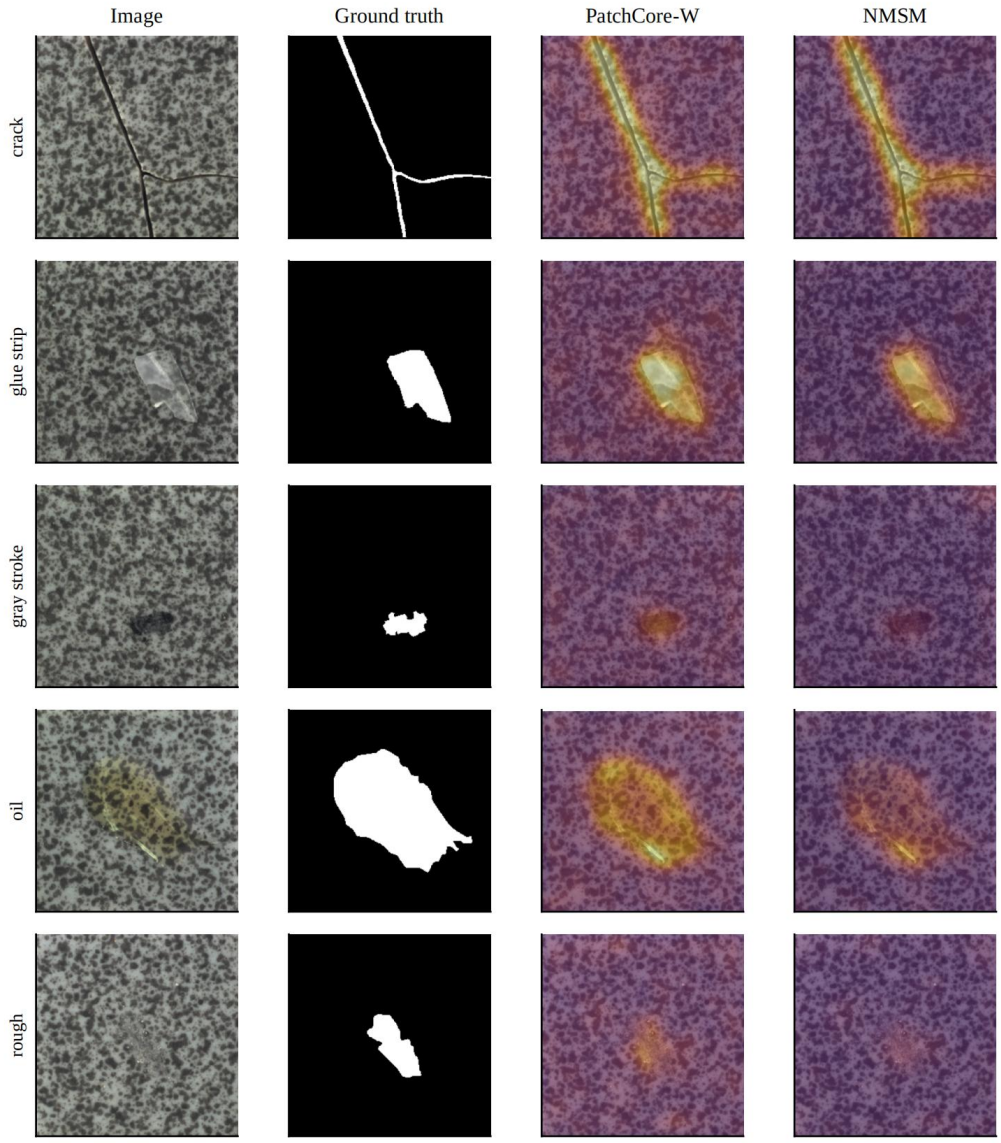

Fig. 6: Fixed Tile examples: four-shot, seed 11, first path per defect type. Image and mask source: MVTec AD [1], CC BY-NC-SA 4.0. This figure, including derived overlays, retains that license.

memory. Robustness to contamination, illumination changes, and cross-device acquisition has not been tested.

Recent few-shot baselines such as AnomalyDINO and PromptAD are not reproduced under the same protocol. No second independent tile dataset, regional AUPRO, or calibrated false-alarm operating point is provided. These omissions limit novelty and practical-advantage claims. This work evaluates established components rather than establishing a universally superior algorithm.

Supplementary diagnostics are post-hoc. They neither erase the initial results nor create an untouched test set. Future structure selection should use separate development data or normal-only proxy objectives before evaluation on data not used for design.

VII. Conclusion

This study evaluates lightweight multiscale memories for few-normal-shot tile anomaly detection under an explicit budget. NMSM improves selected image and pixel AUROC values over a same-backbone PatchCore adaptation, while reducing pixel AP and remaining less accurate than the larger backbone. Supplementary diagnostics do not establish that combining channel normalization and layer balancing outperforms simpler alternatives on the primary localization endpoint. The evidence supports a reproducible account of accuracy–cost tradeoffs and explicit limitations, not comprehensive superiority or production readiness.

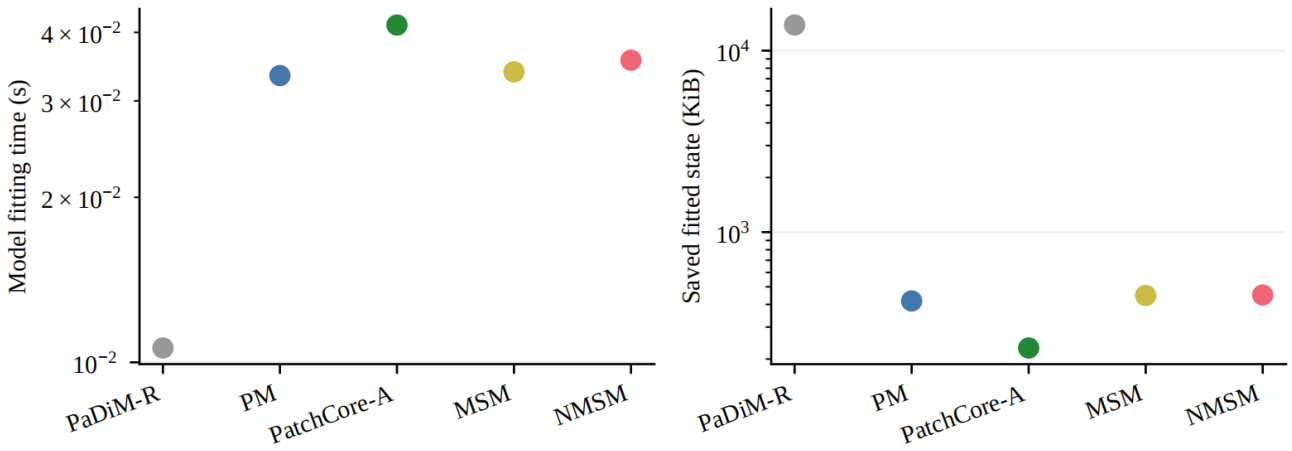

Fig. 7: Mean fitting time excluding backbone extraction and compressed fitted-state size for four-shot Tile. Points are used on logarithmic axes to avoid misleading truncated bar lengths.

References

[1] P. Bergmann, K. Batzner, M. Fauser, *et al.*, "The MVTec Anomaly Detection Dataset: A Comprehensive Real-World Dataset for Unsupervised Anomaly Detection," *International Journal of Computer Vision*, vol. 129, no. 4, 2021, pp. 1038–1059, doi: 10.1007/s11263-020-01400-4.

[2] T. Defard, A. Setkov, A. Loesch, *et al.*, "PaDiM: A Patch Distribution Modeling Framework for Anomaly Detection and Localization," in *Pattern Recognition, ICPR International Workshops and Challenges*, LNCS, 2021, pp. 475–489, doi: 10.1007/978-3-030-68799-1_35.

[3] K. Roth, L. Pemula, J. Zepeda, *et al.*, "Towards Total Recall in Industrial Anomaly Detection," in *Proc. IEEE/CVF Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 14298–14308, doi: 10.1109/cvpr52688.2022.01392.

[4] C.-L. Li, K. Sohn, J. Yoon, *et al.*, "CutPaste: Self-Supervised Learning for Anomaly Detection and Localization," in *Proc. IEEE/CVF Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 9659–9669, doi: 10.1109/cvpr46437.2021.00954.

[5] V. Zavrtanik, M. Kristan, and D. Skocaj, "DRAEM - A Discriminatively Trained Reconstruction Embedding for Surface Anomaly Detection," in *Proc. IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 8310–8319, doi: 10.1109/iccv48922.2021.00822.

[6] G. Wang, S. Han, E. Ding, *et al.*, "Student-Teacher Feature Pyramid Matching for Anomaly Detection," in *Proc. British Machine Vision Conference (BMVC)*, 2021, paper 349, doi: 10.5244/c.35.349.

[7] M. Rudolph, B. Wandt, and B. Rosenhahn, "Same Same but DifferNet: Semi-Supervised Defect Detection With Normalizing Flows," in *Proc. IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2021, pp. 1906–1915, doi: 10.1109/wacv48630.2021.00195.

[8] D. Gudovskiy, S. Ishizaka, and K. Kozuka, "CFLOW-AD: Real-Time Unsupervised Anomaly Detection With Localization via Conditional Normalizing Flows," in *Proc. IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2022, pp. 1819–1828, doi: 10.1109/wacv51458.2022.00188.

[9] H. Deng, and X. Li, "Anomaly Detection via Reverse Distillation From One-Class Embedding," in *Proc. IEEE/CVF Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 9727–9736, doi: 10.1109/cvpr52688.2022.00951.

[10] Z. Liu, Y. Zhou, Y. Xu, *et al.*, "SimpleNet: A Simple Network for Image Anomaly Detection and Localization," in *Proc. IEEE/CVF Computer Vision and Pattern Recognition (CVPR)*, 2023, pp. 20402–20411, doi: 10.1109/cvpr52729.2023.01954.

[11] X. Zhang, S. Li, X. Li, *et al.*, "DeSTSeg: Segmentation Guided Denoising Student-Teacher for Anomaly Detection," in *Proc. IEEE/CVF Computer Vision and Pattern Recognition (CVPR)*, 2023, pp. 3914–3923, doi: 10.1109/cvpr52729.2023.00381.

[12] T. D. Tien, A. T. Nguyen, N. H. Tran, *et al.*, "Revisiting Reverse Distillation for Anomaly Detection," in *Proc. IEEE/CVF Computer Vision and Pattern Recognition (CVPR)*, 2023, pp. 24511–24520, doi: 10.1109/cvpr52729.2023.02348.

[13] J. Jeong, Y. Zou, T. Kim, *et al.*, "WinCLIP: Zero-/Few-Shot Anomaly Classification and Segmentation," in *Proc. IEEE/CVF Computer Vision and Pattern Recognition (CVPR)*, 2023, pp. 19606–19616, doi: 10.1109/cvpr52729.2023.01878.

[14] X. Li, Z. Zhang, X. Tan, *et al.*, "PromptAD: Learning Prompts with only Normal Samples for Few-Shot Anomaly Detection," in *Proc. IEEE/CVF Computer Vision and Pattern Recognition (CVPR)*, 2024, pp. 16848–16858, doi: 10.1109/cvpr52733.2024.01594.

[15] Q. Zhou, G. Pang, Y. Tian, *et al.*, "AnomalyCLIP: Object-agnostic Prompt Learning for Zero-shot Anomaly Detection," in *Proc. International Conference on Learning Representations (ICLR)*, 2024. [Online]. Available: https://openreview.net/forum?id=buC4E91xZE, preprint DOI: 10.48550/arXiv.2310.18961.

[16] S. Damm, M. Laszkiewicz, J. Lederer, *et al.*, "AnomalyDINO: Boosting Patch-Based Few-Shot Anomaly Detection with DINOv2," in *Proc. IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2025, pp. 1319–1329, doi: 10.1109/wacv61041.2025.00136.

[17] Z. Gu, B. Zhu, G. Zhu, *et al.*, "UniVAD: A Training-free Unified Model for Few-shot Visual Anomaly Detection," in *Proc. IEEE/CVF Computer Vision and Pattern Recognition (CVPR)*, 2025, pp. 15194–15203, doi: 10.1109/cvpr52734.2025.01415.

[18] S. Wu, Y. Wang, X. Liu, *et al.*, "DFM: Differentiable Feature Matching for Anomaly Detection," in *Proc. IEEE/CVF Computer Vision and Pattern Recognition (CVPR)*, 2025, pp. 15224–15233, doi: 10.1109/cvpr52734.2025.01418.

[19] K. Batzner, L. Heckler, and R. König, "EfficientAD: Accurate Visual Anomaly Detection at Millisecond-Level Latencies," in *Proc. IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2024, pp. 127–137, doi: 10.1109/wacv57701.2024.00020.

[20] X. Zhang, M. Xu, and X. Zhou, "RealNet: A Feature Selection Network with Realistic Synthetic Anomaly for Anomaly Detection," in *Proc. IEEE/CVF Computer Vision and Pattern Recognition (CVPR)*, 2024, pp. 16699–16708, doi: 10.1109/cvpr52733.2024.01580.

[21] J. Santos, T. Tran, and O. Rippel, "Optimizing PatchCore for Few/many-shot Anomaly Detection," *arXiv preprint arXiv:2307.10792*, 2023, preprint DOI: 10.48550/arXiv.2307.10792.

[22] M. Fučka, V. Zavrtanik, and D. Skočaj, "ObjectCore - Efficient Few-shot Logical Anomaly Detection using Object Representations," in *Proc. IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2026, pp. 3857–3867, doi: 10.1109/wacv61042.2026.00376.