COMPUTATION-PERFORMANCE BENCHMARK FOR GENERALIZED INDUSTRIAL ANOMALY DETECTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Recent industrial anomaly detection (IAD) research increasingly targets three generalization settings: (1) multi-class unified detection with a single shared model, (2) few-shot detection with limited normal data, and (3) zero-shot detection without a designated target-domain support set before deployment. These settings require models to generalize robustly under data-scarce, cross-scenario conditions. Although recent methods achieve high accuracy on benchmark datasets, substantial differences remain in deployment cost. Here, we present a computational taxonomy and a benchmark of 30 representative methods on MVTec-AD and VisA dataset. Evaluation on a single NVIDIA A100 combines detection and localization metrics with batch-one throughput, peak GPU memory, parameter counts, and profiled query FLOPs. Through analyses of predictive performance, few-shot stability, and computational cost, we characterize the strengths and limitations of existing methods under different generalization settings. Our findings reveal that accuracy rankings alone provide an incomplete account of practical effectiveness, motivating an evaluation perspective that connects detection quality with the state and computation required at deployment.

1 INTRODUCTION

Industrial anomaly detection (IAD) is widely used for automated visual inspection in manufacturing, where it can improve production efficiency and reduce the labor required for large-scale quality control. Because real anomalous training data are scarce and expensive to collect, recent IAD methods increasingly learn without target-domain anomaly labels (Pang et al., 2021; Ruff et al., 2021).

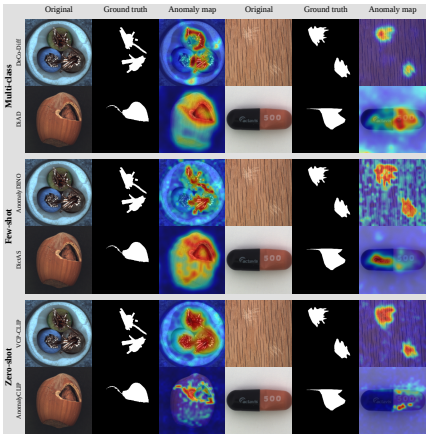


Figure 1: Qualitative localization results in the multi-class, few-shot, and zero-shot settings.

Three generalization settings have become central to recent IAD research and define the scope of this work. Multi-class unified detection trains one shared model on normal images from all categories, replacing a collection of category-specific detectors (You et al., 2022). Few-shot detection receives only a small target-normal support set and must convert those examples into reference state or lightweight adaptation (Jeong et al., 2023; Damm et al., 2025). Zero-shot detection assumes that no designated target-domain normal support set is available before deployment; methods instead rely on pretrained priors, auxiliary source-domain data, or method-specific test-time adaptation (Zhou et al., 2024; Ma et al., 2025; Lu et al., 2026).

Data availability does not determine deployment cost by itself; the scoring path determines what a method stores and how often it processes the query. Some compare test features with learned reconstructions or stored normal references (Lu et al., 2023; Roth et al., 2022; Qu et al., 2025; Lendering et al., 2026); others use iterative generative inference or transfer pretrained visual and

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

semantic representations (He et al., 2024b; Beizaee et al., 2025; Gao et al., 2026). These choices affect both the computation required for each prediction and the state retained during inference.

Accuracy alone is insufficient to determine whether a detector meets a given deployment budget. Existing benchmarks standardize important parts of IAD accuracy evaluation, but have limit in comparisons of deployment among recent multi-class, few-shot, and zero-shot setting. Four questions guide the analysis: how accuracy and cost rankings differ, whether image-level detection reflects localization quality, whether small few-shot advantages persist across support sets, and which deployment costs are impacted by parameter totals.

Figure 1 illustrates visualization outputs across the three settings. Our contributions are shown as following. (i) We organize generalized IAD methods by their anomaly-scoring mechanisms to connect their inference procedures with computational requirements. (ii) We evaluate 30 documented configurations on MVTec-AD and VisA, reporting detection and localization quality together with batch-one efficiency on a common GPU. (iii) We analyze computation-accuracy trade-offs and few-shot ranking stability using five $K = 1$ runs with shared support manifests. A separate single-run $1/2/4$-shot sweep shows how support size affects retained state and memory.

2 Related Work

We group IAD methods by the computation used to produce the anomaly score, because it determines both repeated inference work and retained deployment state. The groups are descriptive rather than exclusive, as recent methods often combine reconstruction, reference comparison, and pretrained alignment.

Reconstruction-residual scoring. Reconstruction methods detect anomalies through the discrepancy between an input and its estimated normal representation. Early approaches reconstruct pixels with autoencoders, variational autoencoders, or adversarial models (Schlegl et al., 2019; Dehaene et al., 2020), whereas recent generalized IAD methods increasingly reconstruct pretrained features. UniAD uses a layer-wise query decoder with neighbor-masked attention, and HVQ-Trans constrains reconstruction through hierarchical vector-quantized normal prototypes (You et al., 2022; Lu et al., 2023). Dinomaly adopts a minimal Transformer architecture with a noisy bottleneck, linear attention, and loose feature reconstruction (Guo et al., 2025). UniNet instead follows a student–teacher design with domain-related feature selection and contrastive objectives (Wei et al., 2025). INP-Former extracts intrinsic normal prototypes from the test image and uses them to guide normal-token reconstruction (Luo et al., 2025). Despite these differences, their released configurations score each query through a fixed feed-forward feature path rather than a dataset-scale retrieval process.

Synthetic-anomaly supervision provides reconstruction or segmentation targets without real defects. DRAEM couples image reconstruction with discriminative segmentation (Zavrtanik et al., 2021). Diffusion methods modify the inference path more substantially. DiAD connects a semantic-guided branch to a Stable-Diffusion denoising network and reconstructs a normal counterpart through repeated latent denoising (He et al., 2024b; Rombach et al., 2022). DeCo-Diff instead learns a direction of deviation from normality and applies deterministic multi-step reverse correction directly to the test image (Beizaee et al., 2025). Both revisit the latent representation across several reverse steps and therefore occupy a different latency regime from feed-forward reconstruction.

Memory-bank and subspace scoring. Reference-based methods summarize normal-image features and score a query by its distance from the retained representation. PatchCore stores a coreset of pretrained CNN patch features and applies nearest-neighbor matching (Roth et al., 2022). Anomaly-DINO transfers this strategy to DINOv2 features with attention-based foreground extraction, while RAD combines global and local retrieval from a multi-level normal-feature memory (Damm et al., 2025; Oquab et al., 2024; Zhang et al., 2026). SubspaceAD replaces explicit patch retrieval with a PCA subspace fitted to few-shot normal features and uses the projection residual as its score (Lendering et al., 2026). FoundAD learns a lightweight nonlinear projector from normal images and synthetic perturbations, then compares each query feature with its projected counterpart (Zhai et al., 2026). These methods retain normal evidence as a feature bank, fitted subspace, or learned projector, which can produce different memory and matching costs despite similar parameter totals.

Vision–language alignment scoring. Vision–language methods derive anomaly evidence from CLIP representations (Radford et al., 2021), although their scoring procedures differ. WinCLIP combines compositional prompts with image-, patch-, and window-level scoring (Jeong et al., 2023). AnomalyCLIP learns object-agnostic normal and abnormal prompts, while AdaCLIP combines static prompts with image-conditioned dynamic prompts (Zhou et al., 2024; Cao et al., 2024). PromptAD learns prompts from normal-only support (Li et al., 2024). AA-CLIP aligns patch features with anomaly-aware text anchors through residual adapters, AdaptCLIP adds visual, textual, and prompt–query adapters, and MoECLIP routes patches to specialized low-rank experts (Ma et al., 2025; Gao et al., 2026; Park et al., 2026).

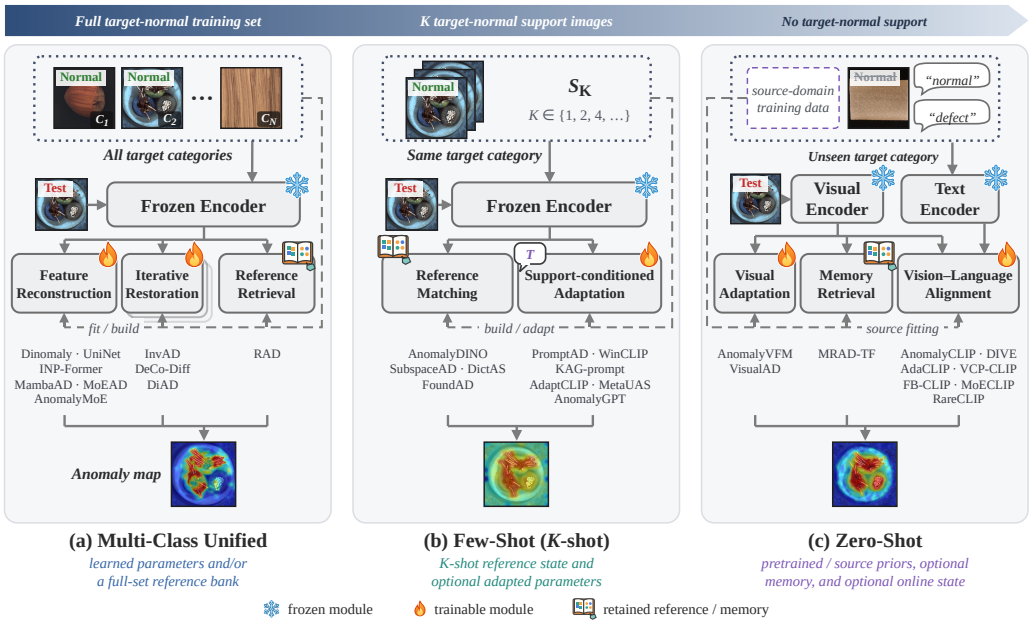
Other methods introduce reference lookup or structured visual reasoning. DictAS constructs a dictionary from normal reference features and identifies regions that cannot be retrieved from it (Qu et al., 2025). KAG-prompt propagates information across a graph of multi-level visual features (Tao et al., 2025). VCP-CLIP injects visual context into prompting, whereas FB-CLIP separates foreground and background evidence (Qu et al., 2024; Hu et al., 2026). MRAD-TF stores image- and pixel-level feature–label pairs from auxiliary data and performs direct similarity retrieval; the broader framework also includes metric-adapted MRAD-FT and prompt-based MRAD-CLIP (Xu et al., 2026). Only MRAD-TF is included in our benchmark. RareCLIP continually updates a prototype memory and rarity statistics from previously processed test images, making its runtime state sequence-dependent (He et al., 2025). DIVE injects text embeddings into shallow and deep visual layers and disentangles anomaly-state features from object semantics (Lu et al., 2026). Thus, cached prompt scoring, window extraction, expert routing, retrieval, and online memory updates can impose substantially different inference costs under the same CLIP backbone.

Other scoring and architectural designs. InvAD performs reconstruction-free anomaly scoring by applying a small number of DDIM inversion steps to pretrained image features and using channel-wise Euclidean norms of the terminal latent representation to form the anomaly map (Sakai et al., 2026). MambaAD combines a pretrained encoder with a multi-scale state-space decoder for long-range and local feature modeling (He et al., 2024a). EfficientAD uses a compact teacher–student predictor together with an autoencoder branch for logical anomalies (Batzner et al., 2024). MoEAD recursively reuses one Transformer block and selects its feed-forward computation from an expert pool, whereas AnomalyMoE assigns patch-, component-, and global-level anomalies to separate reconstruction experts (Meng et al., 2024; Gu et al., 2026).

AnomalyVFM adapts visual foundation models with low-rank feature adapters and synthetic anomaly supervision, while VisualAD removes the text branch and learns normal and anomalous visual tokens in a frozen vision transformer (Fučka et al., 2026; Hou et al., 2026). MetaUAS formulates universal anomaly segmentation as one-prompt visual change segmentation (Gao, 2024). AnomalyGPT augments a large vision–language model with an image decoder and learned prompt interface for localization and language interaction (Gu et al., 2024). UniVAD combines component segmentation, component-aware patch matching, and graph-enhanced component modeling using frozen CLIP and DINOv2 features (Gu et al., 2025).

Anomaly detection benchmarks. Anomalib provides shared training, evaluation, and deployment tools for unsupervised anomaly detectors (Akcay et al., 2022). IM-IAD evaluates 19 algorithms on seven datasets under uniform settings spanning unsupervised, supervised, few-shot, continual, and noisy-label learning, and also reports memory and inference speed (Xie et al., 2024). ADer re-implements 15 methods in a modular multi-class framework and compares them across industrial and medical datasets (Zhang et al., 2025). EfficientAD compares detector latency on common hardware, while MMAD evaluates multimodal large language models on industrial anomaly question answering (Batzner et al., 2024; Jiang et al., 2025).

Question-driven benchmarks in adjacent domains evaluate several properties rather than constructing a single leaderboard. UB-GOLD compares 16 graph-level anomaly and OOD detection methods on 35 datasets across effectiveness, robustness, and efficiency (Wang et al., 2025). Feng et al. (2026) evaluate LLM monitors at a fixed false-positive rate and study OOD detectors and model scaling. Existing IAD benchmarks, however, either re-implement methods in a shared framework, emphasize a particular generalization setting, or predate many recent foundation-model approaches. We instead evaluate 30 documented configurations under multi-class, few-shot, and zero-shot protocols, reporting detection and localization quality, throughput, and peak GPU memory on the same hard-



(a) Multi-Class Unified *learned parameters and/or a full-set reference bank*

(b) Few-Shot (*K*-shot) *K-shot reference state and optional adapted parameters*

(c) Zero-Shot *pretrained / source priors, optional memory, and optional online state*

Figure 2: Computational overview of generalized IAD across the three evaluation settings. Methods are organized by available normal evidence and anomaly scoring mechanisms. Solid arrows indicate query inference; dashed paths indicate offline construction of modules or reference state.

ware. This connects predictive performance to the retained state and repeated computation required by the deployed system.

3 Benchmark Design

We evaluate 30 public implementations, with ten methods in each setting, on MVTec-AD (Bergmann et al., 2019; 2021) and VisA (Zou et al., 2022). Each entry represents a documented configuration. Figure 2 distinguishes the normal evidence available to a method and the computation used to produce its anomaly score. This distinction guides the analysis of model weights, retained reference state, and repeated query processing. Input size and implementation remain part of each configuration.

Detection metrics. We report I-AUROC for image-level detection and P-AUROC and AU-PRO for localization. I-AUROC uses an image score $s_i = g(A_i)$ derived from the anomaly map through each method's aggregation function. P-AUROC is computed by pooling pixel scores and binary ground-truth labels over the evaluated images, rather than averaging image-wise AUROCs.

For AU-PRO, we sweep a shared threshold over all test anomaly maps within each category. At each threshold, PRO averages the fraction of correctly detected pixels in each ground-truth defect region, giving equal weight to all regions. FPR is the fraction of background pixels predicted as anomalous, including pixels from normal images. The area under each category's PRO–FPR curve over $[0, 0.3]$ is divided by 0.3, and the resulting scores are averaged across categories. All scores are reported to three decimal places.

Deployment metrics. All efficiency measurements use a single NVIDIA A100 40GB, batch size one, and five warm-up iterations. We reset the memory counter for every category and macro-average the resulting peaks; the reported memory is therefore a mean of class-level peaks rather than a worst-category requirement. Table 1 then averages the MVTec-AD and VisA measurements. Consecutive remeasurements quantify the repeatability of few-shot efficiency and are independent of the five accuracy runs. Parameter totals exclude reference features and non-parameter buffers. PromptAD is marked † because its recorded time does not include post-processing; we exclude it

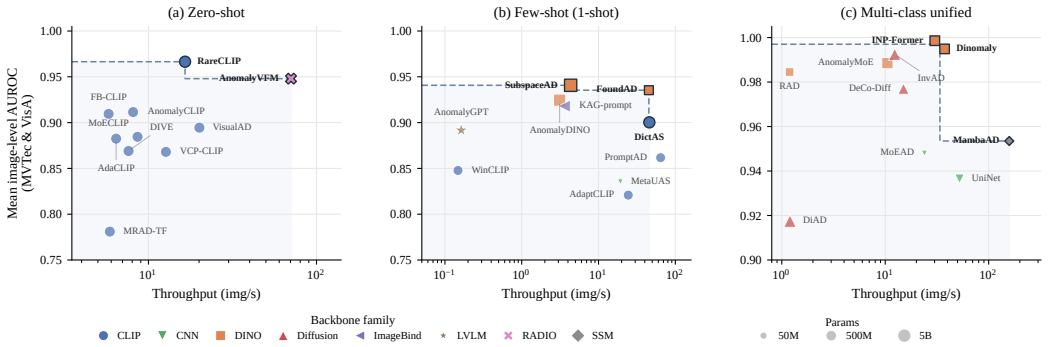


Figure 3: Accuracy–throughput comparisons across the three settings. Values are unweighted means over MVTec-AD and VisA. Marker shape and color indicate backbone family, and size indicates parameter count. PromptAD and MetaUAS are reference points excluded from frontier comparisons because of incomplete timing and unmatched configurations, respectively. Protocol details appear in Appendix B

from throughput comparisons. The exact timing rules, aggregation procedure, and method-specific exceptions appear in Appendix B. Table 3 also reports profiled FLOPs in billions per query for the documented query paths, using the convention that one multiply–accumulate counts as one operation. These shape-based counts do not replace measured throughput or peak memory.

Evaluation settings. Multi-class evaluation uses one detector for all categories. For few-shot evaluation at $K = 1$, all methods use the same frozen support manifests for seeds $\{1, 3, 5, 7, 9\}$; training or adaptation seeds vary with the run where applicable. We first average over categories, then report the mean and sample standard deviation of the five complete runs. A separate single-run $K \in \{1, 2, 4\}$ sweep provides exploratory support-size comparisons without repeated-run uncertainty estimates. Zero-shot methods that require source-domain fitting follow their released cross-dataset recipes. Input size, adaptation, and scoring remain method-specific, so the results compare complete configurations rather than isolated components. SubspaceAD uses the documented 448×448 benchmark input instead of its default 672×672 input.

4 Benchmark Results

Table 1 collects the current measurements and marks unpaired or partial timings. Four patterns matter for interpreting them: the accuracy leader is rarely the cheapest configuration; image-level detection and localization can separate sharply; small few-shot mean gaps need not define a stable ranking; and parameter count does not capture retained state or repeated computation. We compare methods within, rather than across, generalization settings because their access to target-domain normal data differs.

4.1 Comparable Accuracy Can Hide Different Deployment Costs

Similar detection accuracy does not imply similar deployment cost (Fig. 3). In the few-shot setting, FoundAD's MVTec-AD I-AUROC is 0.010 below SubspaceAD and 0.008 below AnomalyDINO. Its two-dataset mean throughput is 45.25 img/s, compared with 4.34 and 3.81 img/s, while its mean peak GPU memory is approximately one tenth of theirs. Accuracy leaders also differ from efficiency leaders in the other settings: the multi-class methods with the highest I-AUROC achieve neither the highest throughput nor the lowest memory, and the most accurate zero-shot configuration is not the fastest.

Frontier comparisons account for the documented timing exceptions. RareCLIP retains its released online target-state updates, so predictions can depend on previously processed test images. PromptAD's timing excludes post-processing. MetaUAS reaches a mean of 19.25 img/s from the remeasured 19.29 and 19.20 img/s runs at 256^2, but its VisA accuracy still requires a matching-input

Table 1: Accuracy and deployment cost of 30 documented configurations. I and P denote image- and pixel-level AUROC. Few-shot accuracy is mean ± sample standard deviation over five $K = 1$ runs. Efficiency values are batch-one means over the two datasets where available; their deviations describe repeated measurements. Memory is in 10^3 MB. Bold indicates the best eligible value within each setting. Method-specific measurement notes appear below.

Method	Venue	MVTec-AD			VisA			Params	Mean efficiency	
		I	P	AU-PRO	I	P	AU-PRO	(M)	(img/s)	(10^3 MB)
Multi-class unified detection										
RAD$^{\diamond}$	ICML'26	0.991	0.985	**0.958**	0.978	0.991	0.946	85.7	1.19	16.914
InvAD	CVPR'26	0.989	0.974	0.928	0.996	**0.998**	**0.983**	1,224.0	12.35	11.772
AnomalyMoE	AAAI'26	0.995	0.981	0.670	0.982	0.990	0.546	539.0	10.50	3.338
UniNet	CVPR'25	0.950	0.959	0.900	0.923	0.983	0.934	153.5	52.28	1.926
Dinomaly	CVPR'25	**0.997**	0.984	0.951	0.996	0.994	0.975	148.0	33.77	0.926
INP-Former	CVPR'25	**0.997**	0.985	0.950	**0.997**	0.996	0.978	155.1	33.49	1.184
DeCo-Diff	CVPR'25	0.992	**0.988**	0.951	0.962	0.983	0.915	478.7	15.02	5.296
MambaAD	NeurIPS'24	0.978	0.974	0.922	0.929	0.984	0.868	25.6	**158.03**	2.105
MoEAD	ECCV'24	0.968	0.970	0.914	0.928	0.986	0.892	**24.3**	23.95	**0.353**
DiAD	AAAI'24	0.967	0.968	0.909	0.868	0.960	0.752	1,561.6	1.20	6.296
Few-shot detection ($K = 1$)										
SubspaceAD	CVPR'26	**0.958** ±**0.009**	**0.975** ±**0.001**	**0.937** < **0.001**	**0.923** ±**0.005**	0.985 < 0.001	0.933 ±0.002	1,136.5	4.34 ±0.02	4.817 < 0.001
FoundAD	ICLR'26	0.948 ±0.010	0.968 ±0.002	0.929 ±0.002	**0.923** ±**0.007**	**0.998** < **0.001**	**0.987** ±**0.002**	96.9	45.25 ±0.23	0.506 < 0.001
AnomalyDINO	WACV'25	0.956 ±0.008	0.949 ±0.002	0.904 ±0.003	0.893 ±0.008	0.965 ±0.001	0.905 ±0.001	1,136.5	3.81 ±0.09	4.495 < 0.001
AdaptCLIP	AAAI'26	0.873 ±0.009	0.952 ±0.004	0.818 ±0.002	0.769 ±0.021	0.924 ±0.014	0.675 ±0.032	209.8	24.40 ±0.10	1.264 < 0.001
DictAS§	ICCV'25	0.940 ±0.005	0.973 ±0.002	0.918 ±0.004	0.861 ±0.029	0.985 ±0.001	0.912 ±0.013	449.3	**49.53** ±**0.08**	1.957 < 0.001
KAG-prompt	AAAI'25	0.950 ±0.008	0.960 ±0.002	0.903 ±0.003	0.886 ±0.016	0.967 ±0.002	0.822 ±0.008	1,284.0	3.58 ±0.06	5.969 < 0.001
MetaUAS$^{\triangle}$	NeurIPS'24	0.907 ±0.007	0.947 ±0.002	0.870 ±0.007	0.764 ±0.019	0.945 ±0.004	0.711 ±0.008	**22.1**	19.25	**0.163** < **0.001**
PromptAD†	CVPR'24	0.911 ±0.009	0.954 ±0.002	0.886 ±0.004	0.813 ±0.022	0.966 ±0.003	0.840 ±0.010	208.4	64.32 ±0.96	0.494 < 0.001
AnomalyGPT*	AAAI'24	0.935 ±0.010	0.954 ±0.003	0.892 ±0.003	0.849 ±0.036	0.967 ±0.003	0.835 ±0.019	8,089.6	0.163 ±0.005	17.558 < 0.001
WinCLIP	CVPR'23	0.905 ±0.012	0.955 ±0.003	0.872 ±0.004	0.790 ±0.019	0.915 ±0.005	0.722 ±0.013	208.4	0.148 ±0.002	0.578 < 0.001
Zero-shot detection										
AnomalyVFM	CVPR'26	0.965	0.920	0.876	0.931	0.947	0.886	**355.4**	**70.50**	5.062
MoECLIP	CVPR'26	0.937	0.915	0.828	0.832	0.947	0.821	432.8	8.61	3.278
FB-CLIP	CVPR'26	0.924	0.919	0.857	0.895	0.963	0.913	433.5	5.79	2.630
VisualAD	CVPR'26	0.922	0.908	0.875	0.867	0.959	0.916	458.7	20.10	**1.931**
MRAD-TF	ICLR'26	0.812	0.867	0.635	0.750	0.910	0.710	427.9	5.89	2.131
RareCLIP‡	ICCV'25	**0.984**	**0.981**	**0.942**	**0.949**	**0.989**	**0.938**	431.9	16.51	2.571
AdaCLIP	ECCV'24	0.900	0.899	0.441	0.865	0.958	0.535	438.6	6.42	2.541
VCP-CLIP	ECCV'24	0.898	0.921	0.875	0.838	0.958	0.904	435.0	12.70	2.647
AnomalyCLIP	ICLR'24	0.935	0.902	0.833	0.888	0.948	0.854	433.6	8.10	2.252
DIVE	ECCV'26	0.901	0.880	0.826	0.837	0.948	0.851	451.1	7.61	2.435

check. We exclude the latter two from the throughput frontier. RAD's plotted timing covers its full online query path, with the bank-transfer treatment specified in Appendix B.

These comparisons make the deployment trade-off explicit: a small difference in detection accuracy can accompany a large difference in memory or throughput. The relevant choice therefore depends on the available resources. We interpret each frontier within its own generalization setting, since the three settings provide different amounts of target-domain normal evidence.

4.2 Image-Level Detection Does Not Ensure Accurate Localization

High image-level accuracy does not guarantee accurate localization. AnomalyMoE, Dinomaly, and INP-Former share DINOv2 ViT-B/14 with registers and the same $448 \rightarrow 392$ input. All achieve at least 0.995 MVTec-AD I-AUROC, but their AU-PRO scores are 0.670, 0.951, and 0.950, respectively. Backbone choice and nominal input size alone therefore do not explain the localization difference.

Pixel- and region-level metrics also provide distinct information. Across the benchmark, five configurations differ by more than 0.150 between mean P-AUROC and mean AU-PRO (Appendix C). P-AUROC measures pixel-wise ranking, whereas AU-PRO measures overlap over connected defect regions along an FPR curve. Their numerical gap does not by itself identify a spatial failure mode, but it shows why the metrics should be reported separately.

The practical implication is to evaluate localization directly when inspection requires locating defects. I-AUROC establishes how well anomalous images are ranked ahead of normal images; it does not assess the corresponding anomaly maps. The shared-backbone comparison motivates closer analysis of decoding, feature aggregation, and scoring, while leaving their individual contributions unresolved.

4.3 Small Few-Shot Gaps Do Not Define Stable Rankings

The five paired $K = 1$ runs show that a small mean advantage need not produce a repeatable ordering. SubspaceAD and AnomalyDINO differ by only 0.002 in mean MVTec-AD I-AUROC, and their order reverses across runs. On VisA, SubspaceAD and FoundAD are tied at the reported I-AUROC precision, with the paired difference again changing sign. Several corresponding AU-PRO comparisons retain the same direction in all five runs. Ranking stability therefore depends on both the dataset and the evaluation metric.

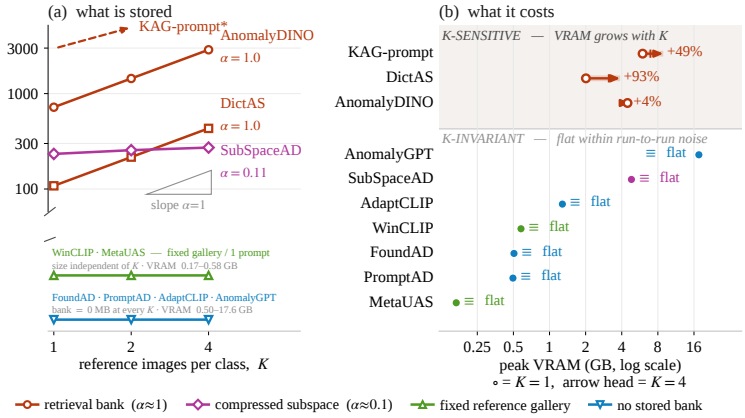


Figure 4: Support-size effects on MVTec-AD: (a) retained feature-bank size and (b) peak GPU memory for $K = 1,2,4$.

Shared support manifests make these comparisons interpretable: each run compares methods using the same normal examples. Individual standard deviations describe each method's variability, whereas paired differences show whether the advantage persists on the same support set. We consequently treat small mean improvements as conditional when their direction changes across runs.

The support-size sweep addresses a separate question: how additional normal examples affect deployment state (Fig. 4). From $K = 1$ to $K = 4$, most observed memory growth is concentrated in methods that retain or expand reference banks. Additional support can therefore affect both the available normal evidence and the state retained at inference. The single-run $K = 2, 4$ measurements describe these configurations but do not establish a general scaling law.

4.4 Deployment Cost Extends beyond Parameter Count

Parameter totals capture only part of deployment cost. MambaAD and MoEAD have similar parameter counts, yet their mean peak GPU memory differs by approximately $6\times$. RAD has a moderate parameter count but the largest multi-class memory footprint, with normal-feature banks and retrieval workspace retained during inference. InvAD instead performs iterative inversion, while DeCo-Diff

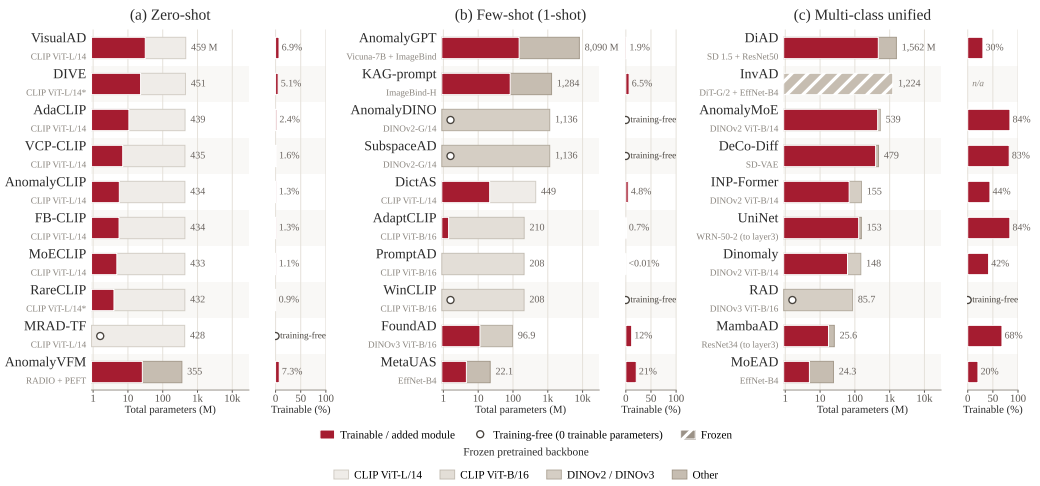


Figure 5: Parameter composition across the three settings. Left panels show total parameters and component breakdowns (M, log scale); right panels show the share of the "Trainable / added module" category, abbreviated as "Trainable (%)". This category includes added weights whose trainability is unresolved. Non-parameter reference state is excluded; component details are provided in Appendix A

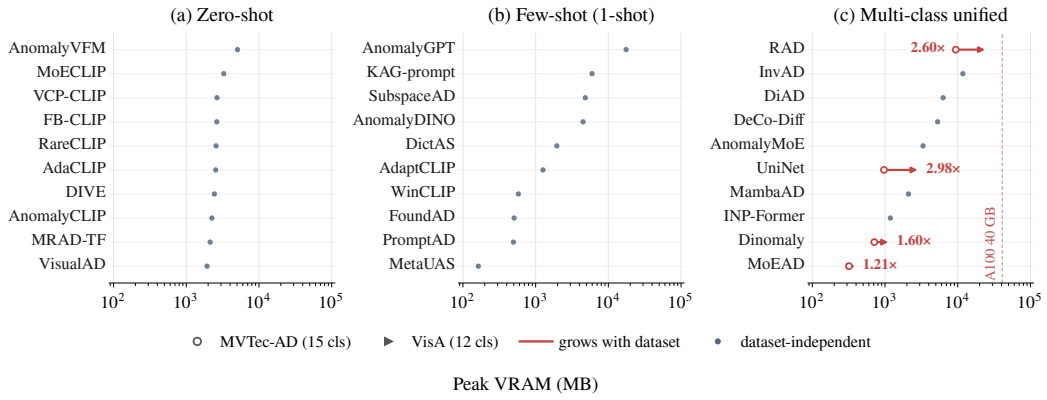


Figure 6: Batch-one peak GPU memory on MVTec-AD and VisA (MB, log scale). Values are averaged over category-level peaks within each dataset; Table 1 reports the two-dataset means. MetaUAS's peak is measured for the 256^2 benchmark configuration.

and DiAD apply repeated correction or denoising steps. These examples distinguish stored model weights, additional inference state, and repeated computation. The parameter–memory correlation also varies across settings (Appendix C).

Comparisons under same backcones. Sharing a backbone does not fix the accuracy–cost operating point. AnomalyMoE, Dinomaly, and INP-Former use DINOv2 ViT-B/14 with registers and the same $448 \rightarrow 392$ input. AnomalyMoE executes 783.6 profiled GFLOPs per query, versus 114.8/107.5 for Dinomaly/INP-Former. Its throughput is roughly one-third as high, while its mean AU-PRO is lower (0.608 versus 0.963/0.964). The frozen DINOv2-G/14 pair SubspaceAD–AnomalyDINO likewise differs in localization despite adding no weights. Within the CLIP ViT-L/14 zero-shot trio at 518×518, VisualAD attains higher mean AU-PRO and throughput with lower mean peak GPU memory than MoECLIP and FB-CLIP, despite slightly more profiled GFLOPs than MoECLIP (542.3 versus 521.0). These comparisons concern complete scoring pipelines, not ablations of individual components.

FLOPs and inference speed. Similar computational workloads can yield different inference speeds (Tables 1 and 3). VisualAD and VCP-CLIP use CLIP ViT-L/14 at 518×518 and require 542.3 and 535.9 GFLOPs per query, respectively. Despite this 1.2% difference, VisualAD achieves $1.58\times$ the mean throughput (20.10 versus 12.70 img/s). The ordering can also reverse across configurations: AnomalyVFM requires approximately twice VisualAD's profiled computation ($1{,}076.3$ GFLOPs), yet runs $3.51\times$ as fast (70.50 img/s). Thus, lower FLOPs do not consistently imply higher throughput. The counts describe profiled arithmetic, while measured throughput also reflects how the query pipeline executes on the GPU. These comparisons do not identify the individual effects of operator implementation, parallelism, or memory access. We therefore report FLOPs and throughput as complementary measures, with profiling coverage and configuration-specific limitations documented in Appendix B.

Figures 5 and 6, together with the query profiles in Table 3, show why parameter count is insufficient as a common cost proxy. Loaded weights contribute to resident memory, reference banks and fitted statistics add state, and iterative inference repeats computation. A parameter total describes the weights but ignores the latter two contributions. Reporting retained state and the query procedure is therefore necessary to interpret measured efficiency.

For an illustrative deployment budget, only PromptAD reaches comparable throughput timing satisfies both a $1{,}000$ MB memory ceiling and a 50 img/s throughput floor under the reported two-dataset averages. The thresholds are examples rather than industrial requirements, and class-averaged peaks do not guarantee that every category fits the memory ceiling. Budget screening should therefore use the measurement scope appropriate to the target deployment.

Scope. The benchmark compares complete documented configurations, with method-specific input resolutions and adaptation procedures. We measure Multi-class and zero-shot accuracy in single runs, and few-shot experiments with five different training or adaptation seeds. Profiled FLOPs cover registered operations on the documented query paths and omit unsupported arithmetic and data movement. Resource measurements do not separately attribute cost to reference preparation, data transfer, scoring, or iterative inference. The results therefore establish observed operating points under common hardware, rather than causal effects of individual components. Further measurements should separate offline preparation from online inference, report mean peak memory, repeat evaluations in different categories, revealing the details under different data distributions

5 Conclusion

This benchmark compares detection quality and deployment cost across 30 documented IAD configurations on MVTec-AD and VisA. Similar detection accuracy can accompany substantially different resource requirements; high image-level AUROC does not ensure comparable localization; and small few-shot advantages can depend on the sampled support set. Parameter totals further omit reference state and repeated computation that affect deployment cost.

These findings support evaluating detectors against the requirements of their intended use. With each generalization setting, memory and throughput constraints determine feasible configurations, localization metrics assess spatial detection quality, and repeated evaluations clarify the reliability of small accuracy differences. Reporting these quantities together provides a more informative basis for method selection than accuracy or parameter count alone.

REFERENCES

Samet Akcay, Dick Ameln, Ashwin Vaidya, Barath Lakshmanan, Nilesh Ahuja, and Utku Genc. Anomalib: A deep learning library for anomaly detection. In *IEEE International Conference on Image Processing (ICIP)*, pp. 1706–1710, 2022.

Kilian Batzner, Lars Heckler, and Rebecca König. EfficientAD: Accurate Visual Anomaly Detection at Millisecond-Level Latencies. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 128–138. IEEE, 2024.

Farzad Beizaee, Gregory A. Lodygensky, Christian Desrosiers, and Jose Dolz. Correcting Deviations from Normality: A Reformulated Diffusion Model for Multi-Class Unsupervised Anomaly Detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 19088–19097. IEEE, 2025.

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 (CVPR)*, pp. 9592–9600. IEEE, 2019.

Paul Bergmann, Kilian Batzner, Michael Fauser, David Sattlegger, and Carsten Steger. The MVTec Anomaly Detection Dataset: A Comprehensive Real-World Dataset for Unsupervised Anomaly Detection. *International Journal of Computer Vision*, 129(4):1038–1059, 2021.

Yunkang Cao, Jiangning Zhang, Luca Frittoli, Yuqi Cheng, Weiming Shen, and Giacomo Boracchi. AdaCLIP: Adapting CLIP with Hybrid Learnable Prompts for Zero-Shot Anomaly Detection. In *Proceedings of the European Conference on Computer Vision (ECCV)*. Springer, 2024.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. Vicuna: An open-source chatbot impressing GPT-4 with 90%* ChatGPT quality. `https://lmsys.org/blog/2023-03-30-vicuna/`, 2023.

Simon Damm, Mike Laszkiewicz, Johannes Lederer, and Asja Fischer. Anomalydino: Boosting patch-based few-shot anomaly detection with dinov2. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 1319–1329, 2025. doi: 10.1109/WACV61041.2025.00136. URL `https://arxiv.org/abs/2405.14529`.

David Dehaene, Oriel Frigo, Sébastien Combrexelle, and Pierre Eline. Iterative Energy-Based Projection on a Normal Data Manifold for Anomaly Localization. In *International Conference on Learning Representations (ICLR)*, 2020.

Dylan Feng, Pragya Srivastava, Anca Dragan, and Cassidy Laidlaw. Benchmarking and improving monitors for out-of-distribution alignment failure in LLMs. *arXiv preprint arXiv:2605.21602*, 2026.

Matic Fučka, Vitjan Zavrtanik, and Danijel Skočaj. AnomalyVFM – Transforming Vision Foundation Models into Zero-Shot Anomaly Detectors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 35555–35566. IEEE, 2026.

Bin-Bin Gao. MetaUAS: Universal Anomaly Segmentation with One-Prompt Meta-Learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 37, pp. 39812–39836, 2024. doi: 10.52202/079017-1257.

Bin-Bin Gao, Yue Zhou, Jiangtao Yan, Yuezhi Cai, Weixi Zhang, Meng Wang, Jun Liu, Yong Liu, Lei Wang, and Chengjie Wang. AdaptCLIP: Adapting CLIP for Universal Visual Anomaly Detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 4095–4103. AAAI Press, 2026. doi: 10.1609/aaai.v40i6.42404.

Rohit Girdhar, Alaaeldin El-Nouby, Zhuang Liu, Mannat Singh, Kalyan Vasudev Alwala, Armand Joulin, and Ishan Misra. ImageBind: One embedding space to bind them all. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

Zhaopeng Gu, Bingke Zhu, Guibo Zhu, Yingying Chen, Ming Tang, and Jinqiao Wang. AnomalyGPT: Detecting Industrial Anomalies Using Large Vision-Language Models. In *Proceedings of the AAAI Conference on Artificial Intelligence*. AAAI Press, 2024.

Zhaopeng Gu, Bingke Zhu, Guibo Zhu, Yingying Chen, Ming Tang, and Jinqiao Wang. UniVAD: A Training-Free Unified Model for Few-Shot Visual Anomaly Detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15194–15203. IEEE, 2025.

Zhaopeng Gu, Bingke Zhu, Guibo Zhu, Yingying Chen, Weifeng Ge, Ming Tang, and Jinqiao Wang. AnomalyMoE: Towards a Language-Free Generalist Model for Unified Visual Anomaly Detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 4348–4356. AAAI Press, 2026. doi: 10.1609/aaai.v40i6.42432.

Jia Guo, Shuai Lu, Weihang Zhang, Fang Chen, Huiqi Li, and Hongen Liao. Dinomaly: The Less Is More Philosophy in Multi-Class Unsupervised Anomaly Detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20405–20415. IEEE, 2025.

Haoyang He, Yuhu Bai, Jiangning Zhang, Qingdong He, Hongxu Chen, Zhenye Gan, Chengjie Wang, Xiangtai Li, Guanzhong Tian, and Lei Xie. MambaAD: Exploring State Space Models for Multi-Class Unsupervised Anomaly Detection. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024a.

Haoyang He, Jiangning Zhang, Hongxu Chen, Xuhai Chen, Zhishan Li, Xu Chen, Yabiao Wang, Chengjie Wang, and Lei Xie. DiAD: A Diffusion-Based Framework for Multi-Class Anomaly Detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 8472–8480. AAAI Press, 2024b.

Jianfang He, Min Cao, Silong Peng, and Qiong Xie. RareCLIP: Rarity-Aware Online Zero-Shot Industrial Anomaly Detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 24478–24487. IEEE, 2025.

Yanning Hou, Peiyuan Li, Zirui Liu, Yitong Wang, Yanran Ruan, Jianfeng Qiu, and Ke Xu. VisualAD: Language-Free Zero-Shot Anomaly Detection via Vision Transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21346–21356. IEEE, 2026.

Ming Hu, Yongsheng Huo, Mingyu Dou, Jianfu Yin, Peng Zhao, Yao Wang, Cong Hu, Bingliang Hu, and Quan Wang. FB-CLIP: Fine-Grained Zero-Shot Anomaly Detection with Foreground-Background Disentanglement. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 35659–35669. IEEE, 2026.

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 (CVPR)*, pp. 19606–19616. IEEE, 2023.

Xi Jiang, Jian Li, Hanqiu Deng, Yong Liu, Bin-Bin Gao, Yifeng Zhou, Jialin Li, Chengjie Wang, and Feng Zheng. MMAD: A comprehensive benchmark for multimodal large language models in industrial anomaly detection. In *International Conference on Learning Representations (ICLR)*, 2025.

Camile Lendering, Erkut Akdag, and Egor Bondarau. SubspaceAD: Training-Free Few-Shot Anomaly Detection via Subspace Modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 28557–28566. IEEE, 2026.

Xurui Li, Jin Li, Biao Leng, Shouhong Zhang, Yongjun Guo, Lei Ma, and Yong Xie. 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 (CVPR)*. IEEE, 2024.

Guanyu Lu, Fang Zhou, and Cheqing Jin. Robust zero-shot anomaly detection under limited auxiliary anomaly priors. In *European Conference on Computer Vision (ECCV)*, 2026. URL `https://arxiv.org/abs/2606.29428`. Accepted.

Ruiying Lu, YuJie Wu, Long Tian, Dandan Wang, Bo Chen, Xiyang Liu, and Ruimin Hu. Hierarchical Vector Quantized Transformer for Multi-Class Unsupervised Anomaly Detection. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36, 2023.

Wei Luo, Yunkang Cao, Haiming Yao, Xiaotian Zhang, Jianan Lou, Yuqi Cheng, Weiming Shen, and Wenyong Yu. Exploring intrinsic normal prototypes within a single image for universal anomaly detection. In *Proceedings of the Computer Vision and Pattern Recognition Conference (CVPR)*, pp. 9974–9983, June 2025.

Wenxin Ma, Xu Zhang, Qingsong Yao, Fenghe Tang, Chenxu Wu, Yingtai Li, Rui Yan, Zihang Jiang, and S. Kevin Zhou. AA-CLIP: Enhancing Zero-Shot Anomaly Detection via Anomaly-Aware CLIP. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4744–4754. IEEE, 2025.

Shiyuan Meng, Wenchao Meng, Qihang Zhou, Shizhong Li, Weiye Hou, and Shibo He. MoEAD: A Parameter-Efficient Model for Multi-Class Anomaly Detection. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 345–361. Springer, 2024.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jégou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning Robust Visual Features without Supervision. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856.

Guansong Pang, Chunhua Shen, Longbing Cao, and Anton van den Hengel. Deep Learning for Anomaly Detection: A Review. *ACM Computing Surveys*, 54(2):1–38, 2021.

Jun Yeong Park, JunYoung Seo, Minji Kang, and Yu Rang Park. MoECLIP: Patch-Specialized Experts for Zero-Shot Anomaly Detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 35534–35544. IEEE, 2026.

William Peebles and Saining Xie. Scalable Diffusion Models with Transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4195–4205. IEEE, 2023.

Zhen Qu, Xian Tao, Mukesh Prasad, Fei Shen, Zhengtao Zhang, Xinyi Gong, and Guiguang Ding. VCP-CLIP: A Visual Context Prompting Model for Zero-Shot Anomaly Segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 301–317. Springer, 2024. doi: 10.1007/978-3-031-72890-7_18.

Zhen Qu, Xian Tao, Xinyi Gong, ShiChen Qu, Xiaopei Zhang, Xingang Wang, Fei Shen, Zhengtao Zhang, Mukesh Prasad, and Guiguang Ding. DictAS: A Framework for Class-Generalizable Few-Shot Anomaly Segmentation via Dictionary Lookup. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 20519–20528. IEEE, 2025.

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. In *Proceedings of the 38th International Conference on Machine Learning*, pp. 8748–8763. PMLR, 2021.

Mike Ranzinger, Greg Heinrich, Jan Kautz, and Pavlo Molchanov. AM-RADIO: Agglomerative vision foundation model – reduce all domains into one. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12490–12500, 2024.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-Resolution Image Synthesis with Latent Diffusion Models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10684–10695. IEEE, 2022.

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 (CVPR)*, pp. 14318–14328. IEEE, 2022.

Lukas Ruff, Jacob R. Kauffmann, Robert A. Vandermeulen, Grégoire Montavon, Wojciech Samek, Marius Kloft, Thomas G. Dietterich, and Klaus-Robert Müller. A Unifying Review of Deep and Shallow Anomaly Detection. *Proceedings of the IEEE*, 109(5):756–795, 2021.

Shunsuke Sakai, Xiangteng He, Chunzhi Gu, Leonid Sigal, and Tatsuhito Hasegawa. InvAD: Inversion-Based Reconstruction-Free Anomaly Detection with Diffusion Models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21389–21398. IEEE, 2026.

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.

Oriane Siméoni, Huy V. Vo, Maximilian Seitzer, et al. DINOv3. *arXiv preprint arXiv:2508.10104*, 2025.

Fenfang Tao, Guo-Sen Xie, Fang Zhao, and Xiangbo Shu. Kernel-Aware Graph Prompt Learning for Few-Shot Anomaly Detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 7347–7355. AAAI Press, 2025. doi: 10.1609/aaai.v39i7.32790.

Yili Wang, Yixin Liu, Xu Shen, Chenyu Li, Rui Miao, Kaize Ding, Ying Wang, Shirui Pan, and Xin Wang. Unifying unsupervised graph-level anomaly detection and out-of-distribution detection: A benchmark. In *International Conference on Learning Representations (ICLR)*, 2025.

Shun Wei, Jielin Jiang, and Xiaolong Xu. UniNet: A Contrastive Learning-Guided Unified Framework with Feature Selection for Anomaly Detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9994–10003. IEEE, 2025.

Guoyang Xie, Jinbao Wang, Jiaqi Liu, Jiayi Lyu, Yong Liu, Chengjie Wang, Feng Zheng, and Yaochu Jin. IM-IAD: Industrial image anomaly detection benchmark in manufacturing. *IEEE Transactions on Cybernetics*, 2024. doi: 10.1109/TCYB.2024.3357213.

Chaoran Xu, Chengkan Lv, Qiyu Chen, Feng Zhang, and Zhengtao Zhang. MRAD: Zero-Shot Anomaly Detection with Memory-Driven Retrieval. In *The Fourteenth International Conference on Learning Representations (ICLR)*, 2026.

Zhiyuan You, Lei Cui, Yujun Shen, Kai Yang, Xin Lu, Yu Zheng, and Xinyi Le. A Unified Model for Multi-Class Anomaly Detection. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, pp. 4571–4584, 2022.

Vitjan Zavrtanik, Matej Kristan, and Danijel Skočaj. DRAEM: A Discriminatively Trained Reconstruction Embedding for Surface Anomaly Detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 8330–8339. IEEE, 2021.

Guangyao Zhai, Yue Zhou, Xinyan Deng, Lars Heckler-Kram, Nassir Navab, and Benjamin Busam. Foundation Visual Encoders Are Secretly Few-Shot Anomaly Detectors. In *The Fourteenth International Conference on Learning Representations (ICLR)*, 2026.

Jiangning Zhang, Haoyang He, Zhenye Gan, Qingdong He, Yuxuan Cai, Zhucun Xue, Yabiao Wang, Chengjie Wang, Lei Xie, and Yong Liu. A Comprehensive Library for Benchmarking Multi-Class Visual Anomaly Detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, pp. 3574–3583. IEEE, 2025.

Xingwu Zhang, Guanxuan Li, Paul Henderson, Gerardo Aragon-Camarasa, and Zijun Long. Is Task-Specific Training Necessary for Anomaly Detection? In *Proceedings of the 43rd International Conference on Machine Learning (ICML)*, volume 306. PMLR, 2026.

Qihang Zhou, Guansong Pang, Yu Tian, Shibo He, and Jiming Chen. AnomalyCLIP: Object-Agnostic Prompt Learning for Zero-Shot Anomaly Detection. In *International Conference on Learning Representations (ICLR)*, 2024.

Yang Zou, Jongheon Jeong, Latha Pemula, Dongqing Zhang, and Onkar Dabeer. SPot-the-Difference Self-Supervised Pre-Training for Anomaly Detection and Segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*. Springer, 2022.

A Parameter Composition

Table 2 separates the shared pretrained or base checkpoint from weights added by each released configuration. MC, FS, and ZS denote the three benchmark settings. Counts are in millions (M). The checkpoint suffix "@336" or "@240" identifies the checkpoint variant, not necessarily the resolution used at inference; input transforms are reported in Table 3.

Table 2: Parameter breakdown by pretrained or base checkpoint. Counts are in millions; reference features and other non-parameter state are excluded. "Added" denotes weights outside the listed base, not necessarily trainable weights.

Method	Setting	Added	Total	Components or retained state
CLIP ViT-L/14@336 (427.94M; VisualAD build: 427.16M)				
MoECLIP	ZS	4.81	432.8	MoE adapter 4.81
FB-CLIP	ZS	5.56	433.5	Prompt learner 5.56
VisualAD	ZS	31.52	458.7	Layer transforms 8.40 + cross-attention 23.12 + tokens/LN 0.004
MRAD-TF	ZS	0	427.9	Training-free retrieval
RareCLIP	ZS	–	431.9	Unassigned difference 3.97; memory-module weights not separately resolved
AdaCLIP	ZS	10.67	438.6	Adapters 10.67
VCP-CLIP	ZS	7.09	435.0	Context prompting 3.94 + four linear layers 3.15
AnomalyCLIP	ZS	5.66	433.6	Prompt learner 5.66
DIVE	ZS	–	451.1	Unassigned difference 23.12; prompts/context and cross-attention not separately resolved
DictAS	FS	21.38	449.3	Value/query/key generators + feature fusion 21.385
CLIP ViT-B/16@240 (208.40M; AdaptCLIP: 208.38M)				
AdaptCLIP	FS	1.41	209.8	Textual 0.015 + visual 0.410 + PQ 0.989 adapters
PromptAD	FS	0.00512	208.4	Prompt learner 0.00512
WinCLIP	FS	0	208.4	Window and reference matching; no added weights
DINOv2 ViT-B/14 with registers (86.58M) (Oquab et al., 2024)				
AnomalyMoE	MC	452.45	539.0	Patch 368.55 + global 81.09 + component 2.61 experts + gating 0.20
Dinomaly	MC	61.42	148.0	Decoder 56.70 + bottleneck 4.72
INP-Former	MC	68.52	155.1	Decoder 56.70 + aggregation 7.09 + bottleneck 4.72 + prototypes 0.005
DINOv2-G/14 (1,136.48M) (Oquab et al., 2024)				
SubspaceAD	FS	0	1,136.5	Fitted PCA state; no added model weights
AnomalyDINO	FS	0	1,136.5	Retained kNN bank; no added model weights
DINOv3 ViT-B/16 (85.70M) (Siméoni et al., 2025)				
RAD	MC	0	85.7	Multi-level feature bank; no added model weights
FoundAD	FS	11.24	96.9	Manifold predictor 11.24
ImageBind-Huge (1,200.79M) (Girdhar et al., 2023)				
KAG-prompt	FS	83.21	1,284.0	KAG adapter 83.21

Method	Setting	Added	Total	Components or retained state
Vicuna-7B (6,738.42M) (Chiang et al., 2023) **+ ImageBind-Huge (1,200.79M)** (Girdhar et al., 2023)				
AnomalyGPT	FS	150.38	8,089.6	LoRA 33.55 + prompt learner 107.38 + image decoder 5.25 + LLaMA projection 4.20
RADIO (+ PEFT), audited base component (329.39M) (Ranzinger et al., 2024)				
AnomalyVFM	ZS	25.96	355.4	Decoder 25.96 + predictor 0.003; PEFT trainability not separately resolved
EfficientNet-B4 (19.34M for MoEAD; 17.55M for MetaUAS)				
MoEAD	MC	4.94	24.3	MoE reconstruction branch 4.94
MetaUAS	FS	4.58	22.1	Decoder + alignment + mask head 4.58
Wide-ResNet50-2 through layer3 (24.86M)				
UniNet	MC	128.60	153.5	Student 24.92 + bottleneck 78.82 + target teacher 24.86 + DFS 0.002
ResNet-34 through layer3 (8.18M)				
MambaAD	MC	17.46	25.6	Mamba student 15.99 + MFF-OCE 1.47
SD-VAE (83.65M) (Rombach et al., 2022)				
DeCo-Diff	MC	394.99	478.7	LDM UNet 394.99; frozen VAE included in total
SD1.5 UNet + CLIP text + SD-VAE + external ResNet-50 (1,089.70M) (Rombach et al., 2022)				
DiAD	MC	471.90	1,561.6	SG control network about 471.87; external ResNet-50 included in total
DiT-G/2 + EfficientNet-B4 (full partition unavailable) (Peebles & Xie, 2023)				
InvAD	MC	–	1,224.0	Denoising head 11.35 independently measured; decoder/full partition unavailable

Counting notes. "Added" refers to weights outside the listed base checkpoint, not necessarily to trainable weights. Figure 5 combines these weights with reported trainable modules under the legend "Trainable / added module"; its "Trainable (%)" axis reports this combined share of total parameters. Unassigned differences from the listed base are included in the displayed share where a finer partition is unavailable. A dash indicates an unresolved partition rather than zero added weights. Rounded components need not sum exactly to the displayed totals. Interpolated positional embeddings, fitted PCA state, and stored reference features are not additional model parameters, although retained state can affect the memory measurements in Table 1.

B Input Configurations and Protocol Notes

Table 3: Input configurations and profiled computation per query. Size denotes resize → crop or patch trim; $s \times s$ denotes direct square resizing. FLOPs are in billions, with one multiply-accumulate counted as one operation. Slash-separated values refer to MVTec-AD/VisA. Configuration-specific notes follow the table.

Setting	Method	Size	FLOPs
Multi-class	RAD	$512 \to 448$	397.9
	InvAD	$256 \times 256 \to$ none	648.3
	AnomalyMoE	$448 \to 392$	783.6
	UniNet	$256 \to 256$	61.2
	Dinomaly	$448 \to 392^{\dagger}$	114.8
	INP-Former	$448 \to 392^{\dagger}$	107.5
	DeCo-Diff	$288 \to 256$	1,043.7
	MambaAD	$256 \to 256$	8.7
	MoEAD	$224 \to 224$ (M); $256 \to 256$ (V)	3.7/4.95
	DiAD	$256 \to 256$	3,245.7
Few-shot	SubspaceAD	$448 \times 448 \to$ none	1,296.3
	FoundAD	$512 \times 512 \to$ none	123.8
	AnomalyDINO	$448^{\S} \to$ patch trim	1,290.8/1,774.5
	AdaptCLIP	$512 \to 512$	176.2
	DictAS	$336 \to 336$	316.9
	KAG-prompt	$224 \to 224$	1,810.2
	MetaUAS	$256 \to 256$ (M); $512 \to 512$ (V, upstream)	10.8/44.3
	PromptAD	$240 \to 240$	30.8
	AnomalyGPT	$224 \times 224 \to$ none	2,871.4
	WinCLIP	$240 \to 240$	3,057.9
Zero-shot	MoECLIP	$518 \to 518$	521.0
	FB-CLIP	$518 \to 518$	710.2
	VisualAD	$518 \times 518^{\ddagger} \to$ none	542.3
	MRAD-TF	$518 \to 518$	636.9
	RareCLIP	$518 \to 518$	803.0
	AdaCLIP	$518 \to 518$	1,339.3
	VCP-CLIP	$518 \to 518$	535.9
	AnomalyCLIP	$518 \to 518$	608.2
	DIVE	$518 \to 518$	628.2
	AnomalyVFM	$768 \times 768 \to$ none	1,076.3

†Default size inferred from the supplied encoder configuration, not passed to the script. ‡VisualAD's size comes from the checkpoint; its CenterCrop(518) after direct 518×518 resizing is a no-op. §AnomalyDINO preserves aspect ratio by resizing the short side to 448, then truncates dimensions to multiples of patch size 14.

Profiled query computation. We use FlopCounterMode to count supported operations on each documented query path, including repeated forward passes but excluding per-category preparation. We report GFLOPs under the convention that one multiply–accumulate counts once. Unregistered operations are omitted, while shape-based formulas need not reflect sparsity or data movement; the values are neither uniform bounds on GPU work nor proxies for latency or energy. In particular, MambaAD's 8.7 GFLOPs does not include its elementwise state-space recurrence.

AnomalyDINO's $1{,}290.8$ GFLOPs covers square MVTec-AD inputs; its VisA value is the 12-class macro mean of profiles at every image's patch-trimmed shape ($1{,}774.5$ GFLOPs). Width is constant within each category but ranges from 476 to 728 across categories. By contrast, two entries remain reference cases: AnomalyGPT's $2{,}871.4$ GFLOPs assumes 128 generated tokens, while timed outputs vary in length; RareCLIP's 803.0 GFLOPs uses a representative online bank state and can change with processing order. Neither supports a fine-grained per-image cost ranking.

MetaUAS's model-only timing excludes image decoding, resizing, and host-to-device transfer, matching the released profiler window. The remeasured 256^2 runs give 19.29 and 19.20 img/s on MVTec-AD and VisA, respectively, with a 0.163×10^3 MB mean peak. The 10.8 GFLOPs profile uses this 256^2 input; the 44.3 GFLOPs VisA value instead represents an upstream 512^2 configuration and should not be paired with the measured throughput. The input provenance of the five-seed VisA accuracy also remains to be checked before making a paired accuracy–efficiency comparison.

Direct square resizing fixes the input tensor shape but can change aspect ratio. By contrast, AnomalyDINO preserves aspect ratio and may process different tensor sizes across images. Table 3 records these input paths, not the method-specific mapping of anomaly maps back to the evaluation masks; we therefore do not attribute localization differences to border handling.

RAD configuration. RAD was evaluated with batch size one on both datasets. We loaded its feature banks one layer at a time and released temporary feature tensors when they were no longer needed. Agreement checks indicated that these memory-saving changes did not alter the detection results. The 397.9 GFLOPs in Table 3 includes five encoder passes. The remeasured query path (encoding, retrieval, and post-processing, but not bank transfers) yields 1.388 img/s on MVTec-AD and 0.987 on VisA, or 1.19 as their unweighted mean in Table 1. With category-amortized bank transfers, the corresponding rates are 1.278 and 0.896 img/s (mean 1.09).

Peak GPU memory was 9.387×10^3 MB on MVTec-AD and 24.441×10^3 MB on VisA, giving the two-dataset mean of 16.914×10^3 MB in Table 1. The RAD paper reports memory footprints of 32.6 GB and 40.7 GB, respectively (Zhang et al., 2026). These figures are not directly comparable: hardware, execution procedure, and memory accounting differ.

C Supplementary Correlation Analysis

Table 4 reports Spearman rank correlations for eligible pairs in Table 1. Memory comparisons exclude MetaUAS; throughput comparisons exclude both MetaUAS and PromptAD. Accuracy and efficiency are averaged equally over MVTec-AD and VisA, with average ranks assigned to ties. These coefficients describe cross-method associations rather than variation across repeated runs.

Table 4: Rank correlations among accuracy and deployment metrics. Spearman's ρ is computed within each setting and over the pooled configurations. Sample counts (MC/FS/ZS/All) are 10/9/10/29 for memory pairs, 10/8/10/28 for throughput pairs, and 10/10/10/30 for accuracy-only pairs.

Quantity A	Quantity B	MC	FS	ZS	All
Parameters	Peak GPU memory	0.515	0.966	−0.479	0.683
Parameters	Throughput	−0.479	−0.395	−0.079	−0.511
Mean I-AUROC	Peak GPU memory	−0.079	0.300	0.261	0.204
Mean I-AUROC	Throughput	0.018	0.286	0.467	0.334
Mean P-AUROC	Mean AU-PRO	0.790	0.855	0.802	0.729

Parameter count is positively correlated with peak GPU memory ($\rho = 0.683$) and negatively correlated with throughput ($\rho = -0.511$) across the pooled configurations. The memory association varies markedly by setting, from 0.966 in few-shot detection to −0.479 in zero-shot detection. Thus, the pooled relationship does not consistently describe individual settings, supporting the configuration-level comparisons in Section 4.

Pixel-level and region-level scores. P-AUROC and AU-PRO yield broadly similar rankings within each setting, but their score magnitudes differ. Across the 30 configurations, the difference between their two-dataset means has a median of 0.070 and an interquartile range of $[0.046, 0.105]$; five differences exceed 0.150. The two metrics therefore provide complementary assessments of localization, although their numerical gap alone does not identify a specific spatial error or its cause.