SubspaceAD: Training-Free Few-Shot Anomaly Detection via Subspace Modeling

Camile Lendering Erkut Akdag Egor Bondarev
AIMS Group, Department of Electrical Engineering, Eindhoven University of Technology
{c.r.lendering, e.akdag, e.bondarev}@tue.nl

Abstract

*Detecting visual anomalies in industrial inspection often requires training with only a few normal images per category. Recent few-shot methods achieve strong results employing foundation-model features, but typically rely on memory banks, auxiliary datasets, or multi-modal tuning of vision-language models. We therefore question whether such complexity is necessary given the feature representations of vision foundation models. To answer this question, we introduce **SubspaceAD**, a training-free method, that operates in two simple stages. First, patch-level features are extracted from a small set of normal images by a frozen DINOv2 backbone. Second, a Principal Component Analysis (PCA) model is fit to these features to estimate the low-dimensional subspace of normal variations. At inference, anomalies are detected via the reconstruction residual with respect to this subspace, producing interpretable and statistically grounded anomaly scores. Despite its simplicity, SubspaceAD achieves state-of-the-art performance across one-shot and few-shot settings without training, prompt tuning, or memory banks. In the one-shot anomaly detection setting, SubspaceAD achieves image-level and pixel-level AUROC of 97.1% and 97.5% on the MVTec-AD dataset, and 93.2% and 98.2% on the VisA dataset, respectively, surpassing prior state-of-the-art results. Code and demo are available at https://github.com/CLendering/SubspaceAD*

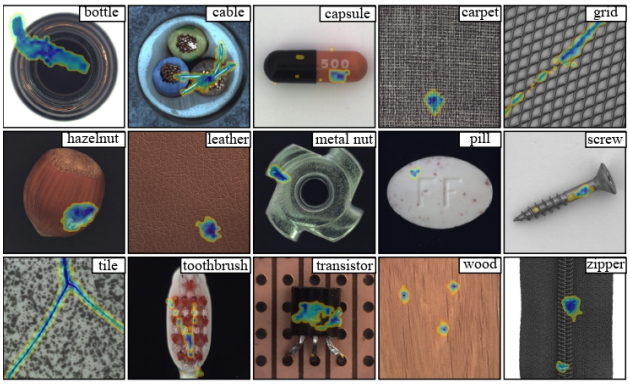


Figure 1. One-shot segmentation results of *SubspaceAD* on the MVTec-AD dataset [4], where *SubspaceAD* only uses one normal image per category. Each example shows a test sample with its predicted anomaly mask (overlaid in dark blue), across all 15 categories of the MVTec-AD dataset.

1. Introduction

Detecting visual anomalies in images is a long-standing challenge in computer vision [7, 28]. In industrial inspection, even subtle deviations from normal appearance, such as scratches, contaminations, or missing components, can lead to downstream failures or safety risks. Development of systems that automatically detect such defects is therefore essential for reliable and cost-efficient production.

The primary challenge in industrial anomaly detection (AD) is data scarcity: full-shot methods require hundreds of defect-free images per category to model normality, which is rarely feasible in practice. At the other extreme, zero-shot methods [18, 40, 42] leverage vision-language models (VLMs) and textual prompts to detect anomalies without any normal samples. However, such methods often struggle with detection of subtle, non-semantic defects *(e.g., small cracks)* that cannot be easily captured by language alone. This paper focuses on the practical and challenging *few-shot* regime, where only a small number of normal images are available to define what constitutes normal appearance for a given object category.

To address the few-shot challenge, recent studies have introduced increasingly complex deep learning techniques, which can be divided into three categories. The first one comprises reconstruction-based approaches [3, 16, 34], which learn to reproduce only normal samples and employ reconstruction residuals as indicators of abnormality. The second category relies on large memory banks of features [10, 11, 31], storing large collections of patch embeddings from normal images and performing anomaly detection via nearest-neighbor retrieval in feature space. More recently, VLM approaches have adapted models like CLIP [30]

through prompt tuning [18, 23, 25] to enable text-guided anomaly detection.

While methods across these three categories achieve strong performance on benchmarks such as MVTec-AD [4] and VisA [43], they have become increasingly complex. These methods often require extensive data augmentation, careful hyperparameter tuning, multi-stage training, auxiliary learning objectives, or large-footprint memory banks, making them difficult to deploy and maintain in real-world industrial settings. In parallel, representation learning has advanced substantially. Foundation vision models, such as DINOv2, produce dense and transferable features that capture both semantic and structural properties of images, even for domains they were never trained on [6, 27, 36]. With features of such quality, one may ask a simple question: *do we still need complex pipelines, large memory banks, and multi-stage tuning to detect anomalies?*

This paper argues that the answer is no. By leveraging strong foundation features, we show that a far simpler alternative is not only feasible but superior. Specifically, we propose a purely statistical approach based on Principal Component Analysis (PCA) [26, 29]. Given only a few normal images, PCA defines a low-dimensional subspace that captures the 'principal' variation of normal appearance. Deviations from this subspace, quantified by the reconstruction residuals, directly indicate anomalies. This approach follows the well-established statistical principle: anomalies (or outliers) manifest as deviations from the dominant PCA subspace of normal data [35].

This minimalist method, termed *SubspaceAD*, is training-free, parameter-light, and interpretable. As demonstrated in Fig. 1, this simple formulation is powerful enough to localize diverse defect patterns even when provided with only a single normal reference sample per category. Extensive experiments show that SubspaceAD surpasses the performance of recently proposed reconstruction-based, memory-bank-based, and VLM-based approaches, suggesting that with sufficiently expressive features, classical statistical modeling can once again serve as a powerful foundation for visual anomaly detection. Summarizing, this paper provides the following contributions:

- We introduce **SubspaceAD**, a minimalist, training-free method for few-shot ($k \in \{1, 2, 4\}$) anomaly detection that combines frozen DINOv2 features with PCA to model normal appearance.
- Through comprehensive evaluations on MVTec-AD and VisA datasets, SubspaceAD outperforms state-of-the-art reconstruction-, memory-bank-, and VLM-based approaches across all few-shot settings.
- SubspaceAD is interpretable and parameter-free, requiring only a single normal image per category and a single forward pass per test image.

2. Related Work

2.1. Reconstruction-Based Approaches

Reconstruction-based approaches detect anomalies by learning to reproduce only normal samples and identifying deviations through reconstruction error. Early methods rely on autoencoders or variational autoencoders to reconstruct normal appearance, assuming that anomalies cannot be accurately recovered [3, 34]. Generative models extend this idea by aligning reconstructions with a learned normal data manifold as demonstrated in [1, 33]. More recent developments introduce perceptual losses, diffusion-based priors, or feature regression strategies to avoid the common pitfall of over-generalization, where the model inadvertently learns to reconstruct anomalous patterns [13, 16]. One of the latest works, FastRecon [15] learns a transform matrix from a few normal samples to reconstruct features as normal, by regression with distribution regularization. While these methods have shown success across industrial defect benchmarks, they require explicit training, hyperparameter tuning, and careful balancing between reconstruction quality and anomaly sensitivity.

2.2. Memory Bank-Based Anomaly Detection

Another major direction in anomaly detection involves storing representative patch features of normal samples in a memory bank and identifying anomalies via nearest-neighbor matching. For instance, SPADE [10] uses multi-resolution feature correspondences inspired by k-NN and operates in a training-free manner, making it suitable for detecting anomalies in few-shot settings. PatchCore [31], another training-free anomaly detection method, reduces memory redundancy by selecting a compact core set of embeddings to improve retrieval efficiency. This method has demonstrated the ability to handle few-shot anomaly detection. Related approaches estimate feature distributions at spatial locations [12], use flow-based transformations for density modeling [41], or distill pre-trained teacher networks to compress normality priors [13]. More recent methods such as AnomalyDINO [11] leverage features from vision foundation models (e.g., DINOv2) to improve both robustness and localization quality. Despite their strong performance, memory-bank methods typically require storing thousands to millions of patch descriptors and performing nearest-neighbor search at inference, which can become computationally heavy, especially in few-shot or multi-category deployment scenarios.

2.3. Foundational Models and VLM-Based Approaches

Large-scale foundation models [8, 17, 21, 24, 30], including vision-only approaches such as, DINO [6, 27], have significantly influenced visual representation learning, both in the unimodal and multimodal domains. With the success of

large-scale vision-language models like CLIP [30], recent works have explored leveraging text prompts for anomaly detection. For instance, WinCLIP [18] is one of the first works to adopt CLIP for anomaly detection. It utilizes manually designed text prompts to detect anomalies across predefined multi-scale windows, while constructing a multi-scale memory bank for feature matching in few-shot settings. Subsequent methods, like AnoVL [14] and PromptAD [23], automate prompt creation or learn prompt adapters, while others attempt to learn generic normality and abnormality prompts across categories using auxiliary datasets [5, 42]. Specifically, PromptAD [23] proposes semantic concatenation to reverse prompt's semantics and directly optimize a set of learnable context vectors. IIPAD [25] instead generates prompts directly from the available normal instances rather than learning category-specific prompts. This enables a single shared prompt space that generalizes across categories, improving few-shot anomaly detection efficiency without extra training data. Although these approaches improve flexibility, they follow a one-class-per-prompt paradigm and often depend on additional normal/anomalous data, prompt tuning, or domain-specific textual priors.

2.4. Few-Shot and Training-Free Anomaly Detection

Few-shot anomaly detection methods vary in how they characterize normal variation. Training-free vision-based approaches, such as DN2 [2], SPADE [10] and PatchCore [32] typically store normal patch features and detect anomalies via nearest-neighbor retrieval. Methods requiring fine-tuning, including PaDiM [12] and GraphCore [39], instead learn parametric models of feature distributions. Beyond purely visual pipelines, vision–language approaches such as ADP [19], WinCLIP [18] and GPT-4V-based anomaly reasoning [40], use text prompts or language alignment to guide few-shot detection, while zero-, few-shot like APRIL-GAN [9] and AnomalyCLIP [42] aim to generalize across categories without additional training. Batched zero-shot frameworks MuSc [22] and ACR [20] further exploit collective statistics across test sets rather than evaluating samples independently. Collectively, these approaches demonstrate a shift toward reducing supervision and eliminating training overhead, while maintaining strong anomaly discrimination. Our work follows this direction but departs from reliance on memory banks or prompt tuning by modeling normal variation through a simple PCA-based subspace formulation.

3. Method

The proposed *SubspaceAD* method models the linear subspace of normal patch features, eliminating the need for memory banks, prompt tuning, or external data. Anomaly scores are computed via reconstruction errors from this subspace, resulting in a training-free, compact, and interpretable method. An overview of *SubspaceAD* is provided in Fig. 2, which operates in two straightforward stages. First, patch-level features are extracted from a small set of k normal images by a frozen DINOv2-G backbone. Second, a Principal Component Analysis (PCA) model is fit to these features to estimate the low-dimensional subspace of normal variation. At inference, anomalies are detected and localized from the reconstruction residuals of test features with respect to this subspace.

3.1. Problem Formulation

Given a small set of k anomaly-free training images $\mathcal{I}_{\text{train}} = \{I_1, \ldots, I_k\}$ and a test image I_{test}, the goal is to define an anomaly scoring function A, which predicts an anomaly likelihood for every spatial position p in I_{test}:

$$A(I_{\text{test}}, p) \in [0, 1]. \tag{1}$$

In the few-shot regime, the key challenge is to model the manifold of normal patch features using only a limited number of clean samples. It is assumed that patch-level features from normal samples lie near a low-dimensional linear subspace embedded in the feature space of a foundation model, while anomalous regions correspond to samples with large reconstruction residuals outside this subspace.

3.2. Feature Extraction

The core of SubspaceAD is the dense feature representation extracted from a pre-trained vision model. A frozen DINOv2-G model [27] is employed as the feature extraction model to obtain patch-level features. Given an input image, the model produces a sequence of patch tokens, where each token corresponds to a 14×14 patch of the image.

Crucially, instead of using only the tokens from the final transformer block, tokens are aggregated from multiple intermediate layers to obtain a more robust representation that balances high-level semantics with low-level spatial detail. This multi-layer fusion improves sensitivity to subtle anomalies while preserving global context cues, a design choice supported in the ablation study (Sec. 4.7, Table 3).

Let $f_l(p) \in \mathbb{R}^D$ denote the patch token at spatial position p from transformer block l, where D is the model's feature dimension (e.g., 1536 for DINOv2-G). Tokens are extracted from a set of layers $\mathcal{L}$ (layers 22–28 in DINOv2-G). The final feature vector $x_p \in \mathbb{R}^D$ for position p is defined as the mean-pooled representation (multi-layer averaging):

$$x_p = \frac{1}{|\mathcal{L}|} \sum_{l \in \mathcal{L}} f_l(p). \tag{2}$$

This process yields a feature vector x_p for each patch. Together, these vectors form a dense feature map $X \in \mathbb{R}^{h \times w \times D}$ for each image, where h and w denote the spatial grid dimensions.

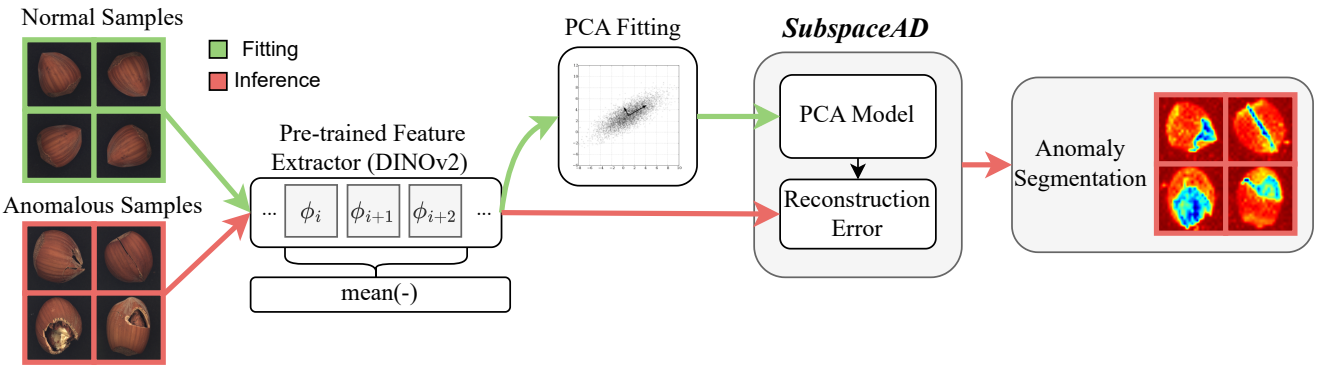


Figure 2. Overview of SubspaceAD. *(Fitting)*: Aggregated patch features are collected from k normal samples using a frozen DINOv2-G model and a PCA model is fitted to capture the subspace of normal variation. *(Inference)*: Features of a test image are extracted, projected onto the normal subspace, and the reconstruction error is computed, providing the anomaly segmentation map directly. PCA figure from [38].

For PCA-based modeling, averaging the features across multiple intermediate layers is particularly beneficial. Since the anomaly score is derived from the residual variance orthogonal to the principal subspace, its reliability depends on how well the feature distribution captures meaningful structure rather than noise. Intermediate layers in DINOv2 contain a mix of semantic and structural information, whereas the deepest layers tend to collapse local detail into category-level abstractions. Therefore, averaging the features across several middle layers stabilizes the covariance estimation, reduces layer-specific variance, and ensures that the principal components represent stable patterns of normal appearance.

To build a representative covariance matrix from only k normal images, data augmentation is applied. For each of the k normal images, $N_a = 30$ augmented views are generated by applying random rotations between $0°$ and $345°$, as rotational variance is common in industrial inspection. Features are extracted from all $k \times (1 + N_a)$ images (the original, plus its augmentations), forming the set of all patch features as X_{normal}. This ensures that the estimated subspace captures common geometric variations and is not biased by a single view. The method consists of a single fitting phase (on k normal images) and an inference phase applied per test image, as illustrated in Fig. 2.

3.3. Subspace Modeling of Normal Features

Normal patch features are modeled via Principal Component Analysis (PCA). This model is fit to the set of all normal features X_{normal}, which contains all patch vectors x_p collected from the k original and augmented normal images (as defined in Sec. 3.2). From this set, the empirical mean $\mu \in \mathbb{R}^D$ and covariance matrix $\Sigma \in \mathbb{R}^{D \times D}$ are computed.

PCA gives a closed-form parameter-free estimate of the dominant linear subspace of the data. We use deterministic PCA for simplicity and numerical stability, making it well-suited for the proposed few-shot regime where overfitting must be avoided. Each patch feature $x \in \mathbb{R}^D$ is modeled as:

$$x = \mu + Cz + \epsilon, \quad z \sim \mathcal{N}(0, I_r), \quad \epsilon \sim \mathcal{N}(0, \sigma^2 I_C), \tag{3}$$

where $C \in \mathbb{R}^{D \times r}$ contains the top r eigenvectors of Σ, $z \in \mathbb{R}^r$ is a latent variable (with I_r being the $r \times r$ identity matrix), and ϵ is an isotropic noise term (with I_C being the $D \times D$ identity matrix). The matrix C forms an orthonormal basis for the subspace of normal variation. Under this probabilistic formulation [37], the squared reconstruction residual $\|(x - \mu) - CC^\top(x - \mu)\|_2^2$ corresponds to the negative log-likelihood component orthogonal to the subspace, thus defining an anomaly score.

The number of retained components r is chosen such that the explained variance exceeds a predefined threshold τ:

$$\sum_{i=1}^{r} \lambda_i \geq \tau \sum_{i=1}^{D} \lambda_i, \qquad \tau = 0.99, \tag{4}$$

where λ_i denotes the i-th eigenvalue of Σ. This high threshold is chosen to ensure the subspace captures the vast majority of normal variation, while discarding minor noise components (see Sec. 4.7 for an empirical analysis). The resulting model is fully described by the mean vector μ and the basis matrix C.

3.4. Anomaly Scoring and Localization

For a test image, its corresponding patch feature map $X_{\text{test}} \in \mathbb{R}^{h \times w \times D}$ is extracted as described in Sec. 3.2. This map consists of all patch feature vectors x_p for the image.

Patch-level Scoring. Each patch feature vector x_p is projected onto the normal subspace:

$$x_{\text{proj}} = \mu + CC^\top(x_p - \mu), \tag{5}$$

and assigned a residual-based anomaly score given by:

$$S(x_p) = \|x_p - x_{\text{proj}}\|_2^2. \tag{6}$$

This score measures the deviation of each feature vector from the principal subspace of normal variation, resulting in a low-resolution anomaly map $M \in \mathbb{R}^{h \times w}$.

Image-level Aggregation. To aggregate patch-level scores into a single image-level prediction, we employ a tail-robust statistic, the empirical tail value-at-risk (TVaR), which averages the top $\rho\%$ of patch scores in the anomaly map M. Let $H_\rho(M)$ denote the set of scores in M at or above the $(100-\rho)$-th percentile. The image-level score s_{img} is then computed as the mean of this set:

$$s_{\text{img}} = \text{mean}\big(H_\rho(M)\big). \tag{7}$$

We set $\rho = 1\%$, following prior work [11], which balances sensitivity to subtle defects with robustness to sparse false positives.

Pixel-level Localization. For visualization and pixel-level evaluation, the patch-level anomaly map M is bilinearly upsampled to the original image resolution and smoothed using a Gaussian filter with $\sigma = 4$ to suppress high-frequency noise while preserving localization accuracy. Finally, the normalized anomaly score function is defined as

$$A(I_{\text{test}}, p) = \text{Norm}\big(S(x_p)\big), \tag{8}$$

where $\text{Norm}(\cdot)$ denotes min–max normalization to $[0, 1]$. These normalized maps are used for visualization, while AUROC and PRO metrics are computed using the raw (unnormalized) scores.

3.5. Complexity and Memory Analysis

Let $n = k \times (1 + N_a) \times (h \times w)$ denote the total number of normal patch features (with k normal training images) and D the feature dimension. PCA fitting requires $O(nD^2)$ time for covariance computation and $O(D^3)$ for eigendecomposition, both negligible in the few-shot regime. The resulting model consists only of $\mu \in \mathbb{R}^D$ and $C \in \mathbb{R}^{D \times r}$, typically requiring less than 1 MB of storage per category. Inference on a 672×672 image takes approximately 300 ms on a single NVIDIA H100 GPU, with the majority of time dominated by the DINOv2-G forward pass (∼226 ms), and the subspace projection and the scoring take only ∼74 ms (see Appendix D for a hardware-normalized analysis).

4. Experiments

The performance of SubspaceAD is evaluated against recent state-of-the-art methods under 1-, 2-, and 4-shot settings, reporting both image-level and pixel-level results (Sec. 4.5). We further assess its generalization under the *batched 0-shot* setting, where the entire unlabeled test set is modeled jointly without any reference images (Sec. 4.6). Finally, ablation studies are conducted to validate the model design choices, including the foundation-model backbone, input image resolution, layer aggregation strategy, and PCA explained-variance threshold τ (Sec. 4.7).

4.1. Datasets

SubspaceAD is evaluated on two widely used industrial anomaly detection benchmarks: MVTec-AD [4] and VisA [43]. Both datasets contain multiple subsets of distinct object and texture categories. MVTec-AD contains 15 categories with image resolutions ranging from 700×700 to 1024×1024 pixels, while VisA includes higher-resolution images (around 1500×1000 pixels) and a broader range of complex real-world anomaly types. Since anomaly detection is formulated as a one-class problem, the training set for each category consists only of normal (defect-free) samples, while the test set contains both normal and anomalous instances. Anomalies in the test set are annotated with ground-truth labels at both the image and pixel levels.

4.2. Evaluation Metrics

Following standard practice [11, 31], performance is evaluated at both the image and pixel levels. Image-level anomaly detection is measured by the Area Under the Receiver Operating Characteristic (AUROC) and Average Precision (AUPR). Pixel-level localization is assessed via pixel-wise AUROC and the Per-Region Overlap (PRO), which accounts for the spatial extent of anomalous regions.

4.3. Implementation Details

The frozen DINOv2-G model [27] is deployed for feature extraction, with features averaged across layers 22–28, as described in Sec. 3.2. For few-shot fitting, $k \in \{1, 2, 4\}$ normal images are randomly selected, and $N_a = 30$ random rotations (from 0° to 345°) are applied to each, except for the orientation-sensitive *transistor* category. For completeness, a fully rotation-agnostic evaluation is provided in Appendix G. The PCA variance threshold is set to $\tau = 0.99$, and TVaR aggregation uses $\rho = 1\%$ [11]. Importantly, we apply a single, fixed image resolution of 672 px across all categories and shots for both MVTec-AD and VisA datasets (see Sec. 4.7 for sensitivity analysis). Each few-shot configuration is evaluated over 5 independent runs with different random seeds, reporting the mean and standard deviation. All experiments are performed on a single NVIDIA H100 GPU.

4.4. Baselines

We compare SubspaceAD against representative few-shot methods across three dominant paradigms: (1) *memory-bank-based* approaches, including SPADE [10], PatchCore [31], and AnomalyDINO [11]; (2) *reconstruction-based* methods, such as FastRecon [15]; and (3) *VLM-based* models, including WinCLIP [18], PromptAD [23], and IIPAD [25].

4.5. Comparison to the State-of-the-Art

Table 1 compares SubspaceAD with recent few-shot methods on MVTec-AD and VisA under 1-, 2-, and 4-shot settings.

Table 1. Comparison of anomaly detection and localization performance on MVTec-AD and VisA across different few-shot settings. Results for SubspaceAD (ours) are reported as mean ± standard deviation over 5 seeds. Best results are in **bold**, and second-best are underlined.

		MVTec-AD				VisA			
		Image-level		Pixel-level		Image-level		Pixel-level	
Setup	**Method**	AUROC	AUPR	AUROC	PRO	AUROC	AUPR	AUROC	PRO
0-shot	WinCLIP	91.8	96.5	85.1	64.6	78.1	81.2	79.6	56.8
1-shot	SPADE	81.0	90.6	91.2	83.9	79.5	82.0	95.6	84.1
	PatchCore	83.4	92.2	92.0	79.7	79.9	82.8	95.4	80.5
	FastRecon	-	-	-	-	-	-	-	-
	WinCLIP	93.1	96.5	95.2	87.1	83.8	85.1	96.4	85.1
	PromptAD	94.6	97.1	95.9	87.9	86.9	88.4	96.7	85.1
	IIPAD	94.2	97.2	96.4	89.8	85.4	87.5	96.9	87.3
	AnomalyDINO	96.6	98.2	96.8	92.7	87.4	89.0	97.8	92.5
	SubspaceAD (ours)	**97.1 ± 0.9**	**98.6 ± 0.4**	**97.5 ± 0.1**	**94.5 ± 0.2**	**93.2 ± 0.8**	**93.0 ± 0.8**	**98.2 ± 0.1**	**95.5 ± 0.1**
2-shot	SPADE	82.9	91.7	92.0	85.7	80.7	82.3	96.2	85.7
	PatchCore	86.3	93.8	93.3	82.3	81.6	84.8	96.1	82.6
	FastRecon	91.0	-	95.9	-	-	-	-	-
	WinCLIP	94.4	97.0	96.0	88.4	84.6	85.8	96.8	86.2
	PromptAD	95.7	97.9	96.2	88.5	88.3	90.0	97.1	85.8
	IIPAD	95.7	97.9	96.7	90.3	86.7	88.6	97.2	87.9
	AnomalyDINO	96.9	98.2	97.0	93.1	89.7	90.7	98.0	93.4
	SubspaceAD (ours)	**97.5 ± 0.7**	**98.8 ± 0.4**	**97.8 ± 0.1**	**94.9 ± 0.1**	**93.8 ± 0.4**	**93.4 ± 0.7**	**98.3 ± 0.0**	**95.7 ± 0.1**
4-shot	SPADE	84.8	92.5	92.7	87.0	81.7	83.4	96.6	87.3
	PatchCore	88.8	94.5	94.3	84.3	85.3	87.5	96.8	84.9
	FastRecon	94.2	-	97.0	-	-	-	-	-
	WinCLIP	95.2	97.3	96.2	89.0	87.3	88.8	97.2	87.6
	PromptAD	96.6	98.5	96.5	90.5	89.1	90.8	97.4	86.2
	IIPAD	96.1	98.1	97.0	91.2	88.3	89.6	97.4	88.3
	AnomalyDINO	97.7	98.7	97.2	93.4	92.6	92.9	98.2	94.1
	SubspaceAD (ours)	**98.0 ± 0.4**	**99.0 ± 0.2**	**97.9 ± 0.1**	**95.1 ± 0.1**	**94.7 ± 0.2**	**94.3 ± 0.5**	**98.4 ± 0.0**	**96.0 ± 0.1**

Using a 672 px resolution (Sec. 4.7), SubspaceAD consistently achieves new state-of-the-art average performance across all evaluated image- and pixel-level metrics.

On MVTec-AD, SubspaceAD attains 97.1% image-level and 97.5% pixel-level AUROC in the 1-shot setting, outperforming all prior methods. On the more challenging VisA benchmark, SubspaceAD achieves 93.2% image-level AUROC and 95.5% PRO, surpassing the prior state-of-the-art (AnomalyDINO) by 5.8% and 3.0%, respectively.

As the number of reference samples increases, our method maintains its lead across all metrics. In the 4-shot setting, SubspaceAD achieves the highest PRO on MVTec-AD (95.1%) and VisA (96.0%). These results validate our central claim: leveraging strong foundation features with a parameter-free statistical model can outperform more complex state-of-the-art methods. We also report performance under the full-shot setting (see Appendix C).

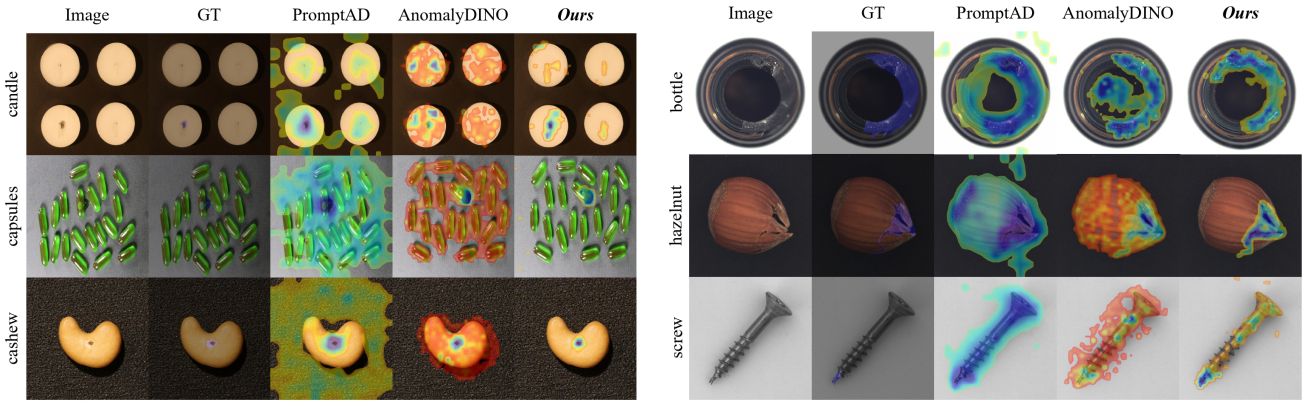
Moreover, qualitative results in Fig. 3 demonstrate that our method produces cleaner, sharper, and more spatially precise anomaly maps across both benchmarks. Further per-category results are included in Appendix A, and representative failure modes are discussed in Appendix F.

4.6. Batched 0-Shot Performance

SubspaceAD is further evaluated under the *batched 0-shot* setting, which differs fundamentally from prompt-based, zero-shot, or few-shot paradigms. Following the protocol of AnomalyDINO [11] and MuSc [22], the entire test set for a category is used to construct the model, assuming that the majority of image patches are anomaly-free. Unlike memory–bank approaches that store all patches across images, SubspaceAD fits a single PCA subspace on all patch tokens extracted from the unlabeled test set of that category and computes anomaly scores based on reconstruction residuals.

Table 2 compares the performance of SubspaceAD against other batched 0-shot methods. SubspaceAD achieves state-of-the-art performance on the VisA dataset with an image-level AUROC of 94.1%, matching the performance of MuSc and outperforming AnomalyDINO. On MVTec-AD, our method achieves a competitive 96.6% AUROC.

SubspaceAD differs from prior batched 0-shot approaches in how it models the unlabeled test set. AnomalyDINO



(a) VisA **(b) MVTec-AD**

Figure 3. Qualitative comparison on VisA and MVTec-AD (1-shot). SubspaceAD produces sharper and more precise anomaly maps than PromptAD [23] and AnomalyDINO [11], with fewer false activations and better alignment with ground-truth defects across both datasets. More qualitative examples are provided in Appendix E.

Table 2. Batched 0-shot anomaly detection. All values are image-level AUROC (%). Best results are in **bold**, second-best are underlined.

Method	MVTec-AD	VisA
Zero-shot methods		
WinCLIP [18]	91.8	78.1
AnomalyCLIP [42]	91.5	82.1
Batched zero-shot methods		
ACR [20]	85.8	–
MuSc [22]	**97.8**	**94.1**
AnomalyDINO-S (448) [11]	93.0	89.7
AnomalyDINO-S (672) [11]	94.2	90.7
SubspaceAD (ours)	96.6	**94.1**

builds a memory bank from all test patches, allowing anomalous regions to retrieve other anomalies as nearest neighbors and suppressing their anomaly scores. MuSc [22] alleviates this through mutual similarity filtering, but it requires dense cross-image comparisons and remains sensitive to contaminated categories. In contrast, SubspaceAD fits a single PCA model to all test tokens, where the principal components capture the shared, high-variance structure of normal data, while rare and uncorrelated anomalies reconstruct poorly and receive high anomaly scores. This compact, distribution-level modeling yields competitive performance on MVTec-AD (96.6%) and achieves state-of-the-art results on VisA (94.1%), showing that even in the batched 0-shot regime, a simple subspace is sufficient for strong anomaly discrimination.

4.7. Ablation Study

We analyze the impact of design choices in SubspaceAD, including (1) input image resolution, (2) layer aggregation strategy, (3) DINOv2 backbone scale, and (4) PCA explained-variance threshold τ. All experiments are performed on the MVTec-AD and VisA datasets.

Image Resolution. Fig. 4 shows the effect of input resolution. On VisA (dashed lines), performance improves from 256 px but quickly saturates, remaining largely stable between 448 px and 672 px. A similar trend is observed on MVTec-AD (solid lines), where all resolutions above 448 px perform comparably across metrics. These results indicate that the method is largely robust to resolution once sufficient spatial detail is available.

Layer Aggregation. Table 3 compares aggregation strategies considering 4-shot results. Using only the last layer leads to a substantial performance drop (89.3% I-AUROC

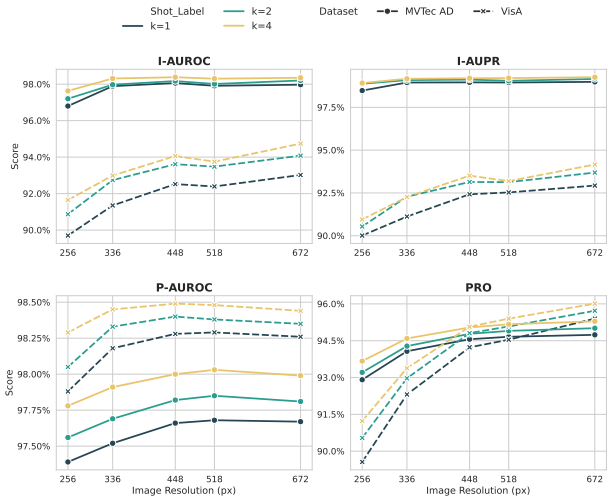


Figure 4. Effect of image resolution on performance across both datasets. Performance peaks at 672 px on both MVTec-AD (solid) and VisA (dashed).

on VisA). Averaging the final 7 layers (34-40) achieves better results but remains suboptimal. The *Mean-pool (Middle-7)* configuration, averaging layers 22-28, delivers the best overall performance, reaching 95.0% PRO on MVTec-AD and 94.1% I-AUROC on VisA. While *Concat (Middle-7)* yields a higher I-AUROC on MVTec-AD (98.6%), our chosen pooling method provides the most robust and discriminative representation across both benchmarks.

Table 3. Evaluation of feature aggregation strategies (4-shot). The selected approach *Mean-pool (Middle-7)* achieves the best overall performance, with image-level AUROC and PRO given in (%).

	MVTec-AD		VisA	
Strategy	I-AUROC	PRO	I-AUROC	PRO
Mean-pool (Middle-7)	98.4	**95.0**	**94.1**	**95.1**
Mean-pool (Final-7)	98.2	93.3	91.9	93.5
Concat (Middle-7)	**98.6**	**95.0**	93.8	95.0
Last layer only	97.6	92.5	89.3	91.2

Backbone Scale. SubspaceAD is evaluated using four DINOv2 backbones: ViT-S/14, ViT-B/14, ViT-L/14, and ViT-G/14, as illustrated in Fig. 5. While larger backbones consistently yield better performance, practical deployment constraints (e.g., edge-device memory and latency) are governed primarily by the chosen encoder. Because SubspaceAD adds negligible computational overhead, it can be readily deployed with smaller, edge-friendly variants like DINOv2-S/14. For completeness, we also report results with DINOv3 backbones (Appendix B), performing slightly worse in this setting.

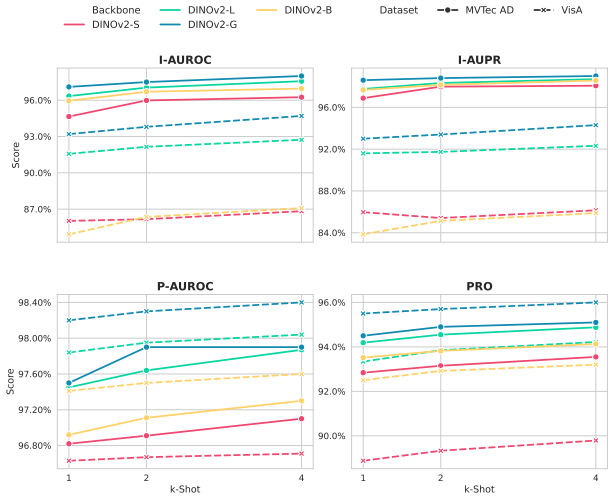


Figure 5. Impact of backbone scale on SubspaceAD performance on both datasets. Performance improves with increasing model capacity, indicating that richer foundation features directly enhance few-shot anomaly detection.

Explained Variance τ. Table 4 shows the effect of the PCA explained-variance threshold τ. Performance remains stable for $\tau \in [0.95, 0.99]$, showing that SubspaceAD is not overly sensitive to this hyperparameter. However, the $\tau = 0.99$ threshold yields the best results for VisA and the strongest localization performance (PRO) on MVTec-AD. Consequently, we adopt $\tau = 0.99$ for all experiments. As expected, setting $\tau = 1.00$ (using the full feature space) causes performance to drop sharply, confirming that anomalies primarily reside in the residual subspace.

Table 4. Analysis of PCA explained variance τ. Image-level AUROC (I-AUROC) and PRO (%) are given for $k \in \{1, 2, 4\}$ on both MVTec-AD and VisA datasets. Best results are in **bold**.

	MVTec-AD			VisA		
τ	k=1	k=2	k=4	k=1	k=2	k=4
I-AUROC (%)						
0.95	98.0	**98.2**	98.4	92.2	93.4	93.7
0.96	**98.1**	**98.2**	98.4	92.4	93.5	93.9
0.97	**98.1**	**98.2**	**98.5**	92.4	**93.6**	93.9
0.99	**98.1**	**98.2**	98.4	**92.5**	**93.6**	**94.1**
1.00	45.6	41.4	40.5	35.3	37.9	38.3
PRO (%)						
0.95	94.4	94.7	94.9	93.9	94.3	94.6
0.96	94.4	94.7	**95.0**	94.0	94.4	94.8
0.97	94.5	94.7	**95.0**	94.1	94.5	94.8
0.99	**94.6**	**94.8**	**95.0**	**94.2**	**94.8**	**95.1**
1.00	55.2	52.7	52.6	48.3	48.6	49.2

Summary. The ablations show that performance is highly dependent on specific design choices. Model scale and the feature aggregation strategy have the strongest influence, followed closely by input resolution. The PCA threshold, in contrast, acts primarily as a fine-tuner. Overall, strong foundation features combined with a simple statistical subspace are sufficient for high-performing few-shot anomaly detection.

5. Conclusion

This paper introduced SubspaceAD, a training-free framework for few-shot visual anomaly detection that leverages the representational strength of vision foundation models. By extracting patch-level features from a frozen DINOv2-G encoder and modeling normal variation through a simple PCA subspace, the method detects anomalies via reconstruction residuals without requiring memory banks, auxiliary datasets, prompt tuning, or any form of training. Despite its simple formulation, SubspaceAD achieves state-of-the-art performance across one- and few-shot settings on both MVTec-AD and VisA datasets, demonstrating that complex architectures and multi-stage optimization are unnecessary when expressive feature representations are available.

Acknowledgements

This work is supported by the ADVISOR ITEA 241007 project.

References

[1] Samet Akcay, Amir Atapour-Abarghouei, and Toby P Breckon. Ganomaly: Semi-supervised anomaly detection via adversarial training. In *Asian conference on computer vision*, pages 622–637. Springer, 2018. 2

[2] Liron Bergman, Niv Cohen, and Yedid Hoshen. Deep nearest neighbor anomaly detection. *arXiv preprint arXiv:2002.10445*, 2020. 3

[3] Paul Bergmann, Sindy Löwe, Michael Fauser, David Sattlegger, and Carsten Steger. Improving unsupervised defect segmentation by applying structural similarity to autoencoders. *arXiv preprint arXiv:1807.02011*, 2018. 1, 2

[4] Paul Bergmann, Michael Fauser, David Sattlegger, and Carsten Steger. Mvtec ad–a comprehensive real-world dataset for unsupervised anomaly detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9592–9600, 2019. 1, 2, 5

[5] 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 *European Conference on Computer Vision*, pages 55–72. Springer, 2024. 3

[6] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9650–9660, 2021. 2

[7] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM computing surveys (CSUR)*, 41(3):1–58, 2009. 1

[8] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PmLR, 2020. 2

[9] Xuhai Chen, Yue Han, and Jiangning Zhang. A zero-/few-shot anomaly classification and segmentation method for cvpr 2023 (vand) workshop challenge tracks 1 &2. *1st Place on Zero-shot AD and 4th Place on Few-shot AD*, 2305:17382, 2023. 3

[10] Niv Cohen and Yedid Hoshen. Sub-image anomaly detection with deep pyramid correspondences. *arXiv preprint arXiv:2005.02357*, 2020. 1, 2, 3, 5

[11] Simon Damm, Mike Laszkiewicz, Johannes Lederer, and Asja Fischer. Anomalydino: Boosting patch-based few-shot anomaly detection with dinov2. In *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 1319–1329. IEEE, 2025. 1, 2, 5, 6, 7, 4

[12] Thomas Defard, Aleksandr Setkov, Angelique Loesch, and Romaric Audigier. Padim: a patch distribution modeling framework for anomaly detection and localization. In *International conference on pattern recognition*, pages 475–489. Springer, 2021. 2, 3

[13] Hanqiu Deng and Xingyu Li. Anomaly detection via reverse distillation from one-class embedding. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9737–9746, 2022. 2

[14] Hanqiu Deng, Zhaoxiang Zhang, Jinan Bao, and Xingyu Li. Anovl: Adapting vision-language models for unified zero-shot anomaly localization. *arXiv preprint arXiv:2308.15939*, 2(5), 2023. 3

[15] Zheng Fang, Xiaoyang Wang, Haocheng Li, Jiejie Liu, Qiugui Hu, and Jimin Xiao. Fastrecon: Few-shot industrial anomaly detection via fast feature reconstruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 17481–17490, 2023. 2, 5

[16] Matic Fučka, Vitjan Zavrtanik, and Danijel Skočaj. Transfusion–a transparency-based diffusion model for anomaly detection. In *European conference on computer vision*, pages 91–108. Springer, 2024. 1, 2

[17] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16000–16009, 2022. 2

[18] Jongheon Jeong, Yang Zou, Taewan Kim, Dongqing Zhang, Avinash Ravichandran, and Onkar Dabeer. Winclip: Zero-/few-shot anomaly classification and segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19606–19616, 2023. 1, 2, 3, 5, 7, 4

[19] Sangkyung Kwak, Jongheon Jeong, Hankook Lee, Woohyuck Kim, Dongho Seo, Woojin Yun, Wonjin Lee, and Jinwoo Shin. Few-shot anomaly detection via personalization. *IEEE Access*, 12:11035–11051, 2024. 3

[20] Aodong Li, Chen Qiu, Marius Kloft, Padhraic Smyth, Maja Rudolph, and Stephan Mandt. Zero-shot anomaly detection via batch normalization. *Advances in Neural Information Processing Systems*, 36:40963–40993, 2023. 3, 7

[21] Chunyuan Li, Zhe Gan, Zhengyuan Yang, Jianwei Yang, Linjie Li, Lijuan Wang, Jianfeng Gao, et al. Multimodal foundation models: From specialists to general-purpose assistants. *Foundations and Trends® in Computer Graphics and Vision*, 16(1-2):1–214, 2024. 2

[22] Xurui Li, Ziming Huang, Feng Xue, and Yu Zhou. Musc: Zero-shot industrial anomaly classification and segmentation with mutual scoring of the unlabeled images. In *The Twelfth International Conference on Learning Representations*, 2024. 3, 6, 7

[23] Xiaofan Li, Zhizhong Zhang, Xin Tan, Chengwei Chen, Yanyun Qu, Yuan Xie, and Lizhuang Ma. Promptad: Learning prompts with only normal samples for few-shot anomaly detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16838–16848, 2024. 2, 3, 5, 7

[24] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. In *European conference on computer vision*, pages 38–55. Springer, 2024. 2

[25] Wenxi Lv, Qinliang Su, and Wenchao Xu. One-for-all few-shot anomaly detection via instance-induced prompt learning. In *The Thirteenth International Conference on Learning Representations*, 2025. 2, 3, 5

[26] Andrzej Maćkiewicz and Waldemar Ratajczak. Principal components analysis (pca). *Computers & Geosciences*, 19 (3):303–342, 1993. 2

[27] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023. 2, 3, 5

[28] Guansong Pang, Chunhua Shen, Longbing Cao, and Anton Van Den Hengel. Deep learning for anomaly detection: A review. *ACM computing surveys (CSUR)*, 54(2):1–38, 2021. 1

[29] Karl Pearson. Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin philosophical magazine and journal of science*, 2(11): 559–572, 1901. 2

[30] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR, 2021. 1, 2, 3

[31] Karsten Roth, Latha Pemula, Joaquin Zepeda, Bernhard Schölkopf, Thomas Brox, and Peter Gehler. Towards total recall in industrial anomaly detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14318–14328, 2022. 1, 2, 5

[32] João Santos, Triet Tran, and Oliver Rippel. Optimizing patchcore for few/many-shot anomaly detection. *arXiv preprint arXiv:2307.10792*, 2023. 3

[33] Thomas Schlegl, Philipp Seeböck, Sebastian M. Waldstein, Ursula Schmidt-Erfurth, and Georg Langs. Unsupervised anomaly detection with generative adversarial networks to guide marker discovery. *CoRR*, abs/1703.05921, 2017. 2

[34] 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. 1, 2

[35] Mei-Ling Shyu, Shu-Ching Chen, Kanoksri Sarinnapakorn, and Liwu Chang. A novel anomaly detection scheme based on principal component classifier. In *Proceedings of International Conference on Data Mining*, 2003. 2

[36] Oriane Siméoni, Huy V Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, et al. Dinov3. *arXiv preprint arXiv:2508.10104*, 2025. 2, 1

[37] Michael E Tipping and Christopher M Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 61(3): 611–622, 1999. 4

[38] Wikipedia contributors. Principal component analysis — Wikipedia, the free encyclopedia, 2026. [Online; accessed 3-March-2026]. 4

[39] Guoyang Xie, Jinbao Wang, Jiaqi Liu, Feng Zheng, and Yaochu Jin. Pushing the limits of fewshot anomaly detection in industry vision: Graphcore. *arXiv preprint arXiv:2301.12082*, 2023. 3

[40] Xiaohao Xu, Yunkang Cao, Huaxin Zhang, Nong Sang, and Xiaonan Huang. Customizing visual-language foundation models for multi-modal anomaly detection and reasoning. *arXiv preprint arXiv:2403.11083*, 2024. 1, 3

[41] Jiawei Yu, Ye Zheng, Xiang Wang, Wei Li, Yushuang Wu, Rui Zhao, and Liwei Wu. Fastflow: Unsupervised anomaly detection and localization via 2d normalizing flows. *arXiv preprint arXiv:2111.07677*, 2021. 2

[42] Qihang Zhou, Guansong Pang, Yu Tian, Shibo He, and Jiming Chen. Anomalyclip: Object-agnostic prompt learning for zero-shot anomaly detection. *arXiv preprint arXiv:2310.18961*, 2023. 1, 3, 7

[43] Yang Zou, Jongheon Jeong, Latha Pemula, Dongqing Zhang, and Onkar Dabeer. Spot-the-difference self-supervised pre-training for anomaly detection and segmentation. In *European conference on computer vision*, pages 392–408. Springer, 2022. 2, 5, 1

SubspaceAD: Training-Free Few-Shot Anomaly Detection via Subspace Modeling

Supplementary Material

A. Per-Category Few-Shot Results

Tables 5 and 6 provide per-category few-shot performance on the MVTec-AD and VisA datasets.

MVTec-AD (Table 5). *SubspaceAD* achieves consistently strong performance across nearly all MVTec-AD categories. Even with only one normal image, several categories (e.g., Bottle, Carpet, Grid, Leather, Tile, Toothbrush) reach perfect or near-perfect image-level AUROC, indicating robust capture of normal appearance from one example. The *Transistor* category is a notable exception, with a 1-shot Pixel PRO score of 68.9%. This limitation is due to the characteristic of patch-based visual anomaly detection, which focuses on local appearance and therefore does not explicitly account for logical or structural anomalies, such as missing or misplaced components.

VisA (Table 6). *SubspaceAD* exhibits greater performance variability across VisA categories, which is expected given the dataset's higher visual diversity and complex backgrounds [43]. Categories such as *Cashew* and *Chewing Gum* achieve excellent results, with 1-shot image-level AUROC scores of 97.7% and 99.2%, respectively. Conversely, categories like *Macaroni2* (80.4%) and *PCB3* (86.5%) are more challenging. This degradation primarily stems from background artifacts that resemble true defects and from the high intra-class variability of normal samples. Both factors hinder patch-based methods from forming a compact subspace of normality from only a few shots.

B. Performance of DINOv3 Backbones

For completeness, we report the few-shot results using the DINOv3-7B backbone [36] in Table 7. These results are obtained using 4096-dimensional features, extracted from layers 22–28 with 16×16 patch tokens. SubspaceAD$_{448}$ and SubspaceAD$_{672}$ denote models evaluated at input image resolutions of 448×448 and 672×672 pixels, respectively. As shown in Table 7, the DINOv3 backbone does not improve over the DINOv2-G backbone used in the main paper (see Table 1), with DINOv2-G remaining stronger overall, especially for localization metrics.

C. Full-Shot Setting Analysis

A comparison in the full-shot setting, where all available normal training samples are used to build the model of normality, is provided in Table 8. On MVTec-AD, AnomalyDINO obtains slightly higher image-level performance, with 99.5% I-AUROC compared to 99.2% for *SubspaceAD*. Both methods achieve the same Pixel AUROC of 98.2%, while *SubspaceAD* obtains a higher Pixel PRO score of 95.6% compared to 95.0%.
On the more challenging VisA dataset, *SubspaceAD* surpasses AnomalyDINO across all reported metrics, including image-level AUROC (98.2% vs. 97.6%), image-level AUPR (98.3% vs. 98.0%), pixel-level AUROC (99.1% vs. 98.8%), and Pixel PRO (96.9% vs. 96.1%). This is notable because AnomalyDINO stores dense patch-level features for every normal image and performs K-NN retrieval over these embeddings, typically on the order of 10^6 feature vectors per category on VisA. In contrast, *SubspaceAD* performs a single forward pass and a lightweight subspace projection, requiring no feature memory banks.

D. Inference Time Analysis

An analysis of inference speed is presented in Table 10. To ensure an algorithmic comparison independent of hardware, we further evaluate the standalone *scoring head* latency (isolated from the backbone forward pass) in Table 9.

End-to-End Latency. Table 10 reports the end-to-end inference times for various backbone configurations. Because these measurements were obtained on different hardware (NVIDIA A40 for AnomalyDINO vs. NVIDIA H100 for SubspaceAD), direct wall-clock speed comparisons are not appropriate. For both methods, the total inference time is primarily dominated by the forward pass of the frozen backbone. The key distinction is architectural: SubspaceAD achieves its performance using only the backbone and a lightweight projection, eliminating the need to store or search through feature memory banks.

Algorithmic Fairness (Scoring Head Only). To isolate the scoring mechanism from the backbone's overhead, we measured the time required to compute anomaly scores from extracted features on an H100. As shown in Table 9, both methods exhibit comparable latency in the few-shot regime. SubspaceAD requires ≈74 ms per image, while AnomalyDINO requires ≈80 ms. While our subspace projection time is strictly invariant to the size of the support set once the PCA components are determined, the k-NN retrieval overhead of AnomalyDINO remains similarly marginal for the small values of K evaluated here.

E. Additional Qualitative Results

To complement the qualitative experiments in the main paper (Fig. 1 and Fig. 3), this section provides a more extensive set of qualitative results. We present detailed anomaly maps for representative samples from the VisA dataset in Fig. 6 and the MVTec-AD dataset in Fig. 7. These examples further illustrate the model's localization performance across diverse object categories and anomaly types.

F. Failure Cases and Limitations

While SubspaceAD demonstrates strong performance, it is important to acknowledge its limitations, particularly those inherent to patch-based, few-shot methodologies. Two primary modes of failure are identified, which are common challenges in this domain and are illustrated in Fig. 8.

Table 5. Detailed few-shot anomaly segmentation results of *SubspaceAD* on the MVTec AD dataset. We report mean Image AUROC (%), Image AUPR (%), Pixel AUROC (%), and Pixel PRO (%) results.

Category	1-shot				2-shot				4-shot			
	Img AUROC	Img AUPR	Pxl AUROC	PRO	Img AUROC	Img AUPR	Pxl AUROC	PRO	Img AUROC	Img AUPR	Pxl AUROC	PRO
Bottle	100.0 ± 0.0	100.0 ± 0.0	98.6 ± 0.0	96.5 ± 0.1	100.0 ± 0.0	100.0 ± 0.0	98.7 ± 0.0	96.7 ± 0.1	100.0 ± 0.0	100.0 ± 0.0	98.7 ± 0.0	96.7 ± 0.1
Cable	92.1 ± 1.4	95.5 ± 0.7	95.6 ± 0.2	90.3 ± 0.7	92.7 ± 1.2	96.2 ± 0.6	95.6 ± 0.2	90.6 ± 0.5	93.5 ± 0.7	96.7 ± 0.5	95.7 ± 0.2	90.9 ± 0.4
Capsule	88.1 ± 11.3	96.8 ± 3.9	98.0 ± 0.2	96.7 ± 0.5	92.7 ± 3.6	98.4 ± 0.9	98.2 ± 0.1	97.1 ± 0.2	93.4 ± 4.8	98.5 ± 1.2	98.3 ± 0.1	97.3 ± 0.2
Carpet	100.0 ± 0.0	100.0 ± 0.0	99.2 ± 0.0	98.2 ± 0.1	100.0 ± 0.0	100.0 ± 0.0	99.2 ± 0.0	98.2 ± 0.0	100.0 ± 0.0	100.0 ± 0.0	99.3 ± 0.0	98.3 ± 0.0
Grid	100.0 ± 0.0	100.0 ± 0.0	99.5 ± 0.0	98.1 ± 0.1	100.0 ± 0.0	100.0 ± 0.0	99.5 ± 0.0	98.2 ± 0.1	100.0 ± 0.0	100.0 ± 0.0	99.5 ± 0.0	98.3 ± 0.1
Hazelnut	97.6 ± 3.4	98.6 ± 2.0	99.5 ± 0.2	97.1 ± 1.2	97.3 ± 6.0	98.1 ± 4.3	99.6 ± 0.1	97.8 ± 0.6	99.4 ± 1.4	99.6 ± 0.8	99.6 ± 0.0	98.1 ± 0.3
Leather	100.0 ± 0.0	100.0 ± 0.0	98.9 ± 0.0	98.3 ± 0.1	100.0 ± 0.0	100.0 ± 0.0	98.9 ± 0.0	98.3 ± 0.1	100.0 ± 0.0	100.0 ± 0.0	98.9 ± 0.0	98.3 ± 0.1
Metal Nut	100.0 ± 0.0	100.0 ± 0.0	97.2 ± 0.9	95.9 ± 0.9	100.0 ± 0.0	100.0 ± 0.0	97.8 ± 0.2	96.5 ± 0.2	100.0 ± 0.0	100.0 ± 0.0	97.8 ± 0.1	96.5 ± 0.2
Pill	96.2 ± 0.8	99.2 ± 0.2	95.5 ± 0.5	97.8 ± 0.1	96.6 ± 0.5	99.4 ± 0.1	95.8 ± 0.2	97.9 ± 0.1	97.2 ± 0.8	99.4 ± 0.2	96.2 ± 0.4	98.1 ± 0.2
Screw	88.4 ± 1.7	95.4 ± 0.9	99.1 ± 0.1	96.2 ± 0.3	90.2 ± 2.8	96.2 ± 1.2	99.2 ± 0.1	96.7 ± 0.3	91.9 ± 1.1	96.9 ± 0.5	99.3 ± 0.0	97.0 ± 0.1
Tile	100.0 ± 0.1	100.0 ± 0.0	97.7 ± 0.1	94.8 ± 0.2	100.0 ± 0.0	100.0 ± 0.0	97.7 ± 0.1	94.7 ± 0.2	100.0 ± 0.1	100.0 ± 0.0	97.7 ± 0.1	94.6 ± 0.1
Toothbrush	98.8 ± 1.4	99.5 ± 0.6	98.9 ± 0.5	97.0 ± 0.6	96.7 ± 3.4	98.7 ± 1.3	98.8 ± 0.5	96.9 ± 0.7	98.8 ± 0.7	99.6 ± 0.3	99.2 ± 0.2	97.5 ± 0.4
Transistor	96.6 ± 0.8	94.5 ± 1.4	90.4 ± 1.1	68.9 ± 1.4	96.7 ± 1.3	94.7 ± 2.3	92.1 ± 0.8	71.5 ± 1.2	96.3 ± 2.4	94.7 ± 3.0	92.6 ± 1.1	72.7 ± 1.5
Wood	99.6 ± 0.2	99.9 ± 0.0	96.7 ± 0.3	96.5 ± 0.2	99.7 ± 0.1	99.9 ± 0.0	96.8 ± 0.5	96.7 ± 0.2	99.7 ± 0.1	99.9 ± 0.0	96.8 ± 0.2	96.6 ± 0.1
Zipper	99.1 ± 0.5	99.8 ± 0.1	98.3 ± 0.2	95.4 ± 0.6	99.5 ± 0.2	99.9 ± 0.0	98.4 ± 0.2	95.8 ± 0.4	99.5 ± 0.2	99.9 ± 0.1	98.6 ± 0.1	96.2 ± 0.3
Mean	**97.1 ± 0.9**	**98.6 ± 0.4**	**97.5 ± 0.1**	**94.5 ± 0.2**	**97.5 ± 0.7**	**98.8 ± 0.4**	**97.8 ± 0.1**	**94.9 ± 0.1**	**98.0 ± 0.4**	**99.0 ± 0.2**	**97.9 ± 0.1**	**95.1 ± 0.1**

Table 6. Detailed few-shot anomaly segmentation results of *SubspaceAD* on the VisA dataset. We report mean Image AUROC (%), Image AUPR (%), Pixel AUROC (%), and Pixel PRO (%) results.

Category	1-shot				2-shot				4-shot			
	Img AUROC	Img AUPR	Pxl AUROC	PRO	Img AUROC	Img AUPR	Pxl AUROC	PRO	Img AUROC	Img AUPR	Pxl AUROC	PRO
Candle	94.4 ± 0.8	94.2 ± 0.7	99.4 ± 0.0	97.9 ± 0.2	94.2 ± 0.5	93.8 ± 0.7	99.4 ± 0.0	98.1 ± 0.1	93.9 ± 0.9	93.4 ± 1.1	99.4 ± 0.0	98.1 ± 0.1
Capsules	96.5 ± 0.7	97.9 ± 0.4	98.5 ± 0.2	97.0 ± 0.3	96.8 ± 0.3	98.1 ± 0.2	98.6 ± 0.1	97.2 ± 0.3	97.0 ± 0.9	98.2 ± 0.6	98.6 ± 0.1	97.3 ± 0.3
Cashew	97.7 ± 0.7	98.9 ± 0.3	99.0 ± 0.0	98.5 ± 0.2	97.9 ± 1.3	99.0 ± 0.6	99.2 ± 0.1	98.6 ± 0.1	98.7 ± 0.8	99.4 ± 0.4	99.3 ± 0.1	98.7 ± 0.1
Chewing Gum	99.2 ± 0.1	99.6 ± 0.1	99.6 ± 0.0	96.0 ± 0.2	99.1 ± 0.1	99.6 ± 0.0	99.6 ± 0.0	96.3 ± 0.1	99.1 ± 0.0	99.5 ± 0.0	99.6 ± 0.0	96.2 ± 0.1
Fryum	97.0 ± 0.6	98.7 ± 0.3	96.5 ± 0.3	95.4 ± 0.5	97.9 ± 0.1	99.1 ± 0.1	96.8 ± 0.1	95.5 ± 0.3	98.1 ± 0.3	99.2 ± 0.1	97.0 ± 0.1	95.7 ± 0.3
Macaroni1	92.0 ± 1.1	91.5 ± 1.4	99.6 ± 0.0	95.8 ± 0.3	92.6 ± 1.4	92.1 ± 1.4	99.7 ± 0.0	96.2 ± 0.4	94.2 ± 1.6	93.7 ± 1.3	99.8 ± 0.0	96.8 ± 0.4
Macaroni2	80.4 ± 5.1	76.2 ± 8.0	99.7 ± 0.0	97.5 ± 0.5	81.4 ± 5.4	77.1 ± 9.0	99.7 ± 0.1	97.8 ± 0.2	84.7 ± 3.9	81.8 ± 6.5	99.8 ± 0.0	98.0 ± 0.2
PCB1	92.4 ± 3.2	90.8 ± 2.6	99.3 ± 0.0	95.0 ± 0.5	92.8 ± 1.4	91.2 ± 1.0	99.3 ± 0.0	95.1 ± 0.3	93.2 ± 0.9	91.6 ± 0.9	99.4 ± 0.0	95.3 ± 0.1
PCB2	88.7 ± 2.2	86.7 ± 2.1	96.9 ± 0.3	91.4 ± 0.6	90.5 ± 1.0	87.5 ± 1.6	97.1 ± 0.1	92.2 ± 0.3	91.7 ± 1.3	88.9 ± 1.7	97.2 ± 0.0	92.7 ± 0.2
PCB3	86.5 ± 1.8	86.1 ± 2.2	95.0 ± 0.4	90.4 ± 0.8	88.0 ± 2.6	87.4 ± 3.1	95.2 ± 0.1	91.1 ± 0.6	89.9 ± 2.3	89.8 ± 2.0	95.4 ± 0.2	91.8 ± 0.4
PCB4	98.3 ± 0.7	98.1 ± 0.8	96.4 ± 0.4	92.2 ± 1.1	98.1 ± 0.6	97.8 ± 0.8	96.6 ± 0.3	92.3 ± 0.9	98.4 ± 0.4	98.0 ± 0.6	96.8 ± 0.2	92.4 ± 1.0
Pipe Fryum	95.7 ± 2.3	97.6 ± 1.3	98.7 ± 0.1	98.3 ± 0.1	96.1 ± 1.9	97.8 ± 1.0	98.9 ± 0.1	98.4 ± 0.1	97.8 ± 0.3	98.7 ± 0.2	99.0 ± 0.0	98.5 ± 0.1
Mean	**93.2 ± 0.8**	**93.0 ± 0.8**	**98.2 ± 0.1**	**95.5 ± 0.1**	**93.8 ± 0.4**	**93.4 ± 0.7**	**98.3 ± 0.0**	**95.7 ± 0.1**	**94.7 ± 0.2**	**94.3 ± 0.5**	**98.4 ± 0.0**	**96.0 ± 0.1**

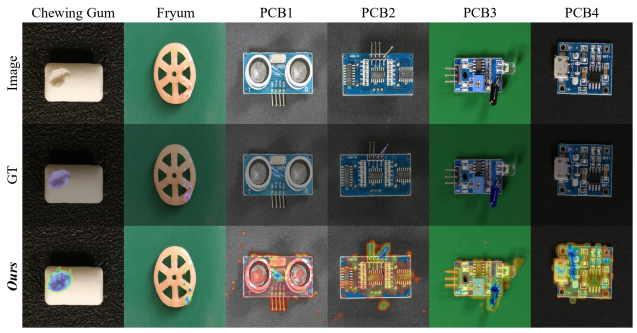


Figure 6. *Additional qualitative results on VisA.* Examples from six categories (*PCB1–4*, *Chewing Gum*, *Fryum*). Rows show the input image, ground-truth mask, and our prediction. *SubspaceAD* accurately localizes both subtle texture anomalies and fine structural defects across diverse VisA domains.

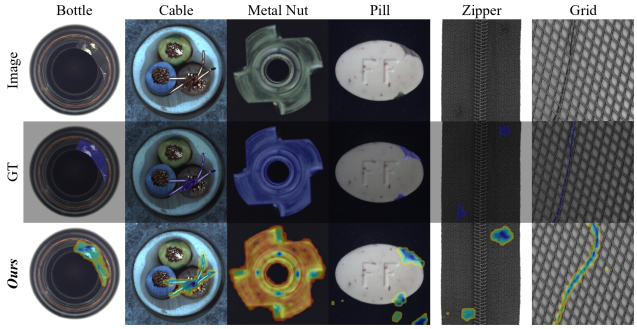


Figure 7. *Additional qualitative results on MVTec-AD.* Examples from six categories (*Bottle*, *Cable*, *Metal Nut*, *Pill*, *Zipper*, *Grid*). Rows show the input image, ground-truth mask, and our prediction. *SubspaceAD* effectively localizes structural, positional, and surface anomalies across varied MVTec categories.

Logical and Structural Anomalies. As a patch-based method, SubspaceAD excels at modeling the *local appearance* and *texture* of normal samples. However, it does not explicitly model global spatial relationships or semantic rules. Consequently, it struggles with logical anomalies, such as a missing component. For example, in the MVTec-AD *Transistor* category (discussed in Appendix A), when a transistor is missing, the exposed circuit-board background may be incorrectly identified as normal texture. The model lacks the semantic, object-level understanding to know that a component *should* be present in that specific location.

Background Artifacts and High Intra-Class Variance. The model struggles in categories with high normal variance or complex, cluttered backgrounds, as seen in some parts of the VisA dataset. As noted in Appendix A, categories Macaroni2 and PCB3

Table 7. Few-shot anomaly detection and localization results using the DINOv3-7B backbone. This backbone is consistently outperformed by DINOv2-G (see Table 1 in the main manuscript).

		MVTec-AD				VisA			
		Image-level		Pixel-level		Image-level		Pixel-level	
Setup	**Method**	AUROC	AUPR	AUROC	PRO	AUROC	AUPR	AUROC	PRO
1-shot	SubspaceAD$_{448}$ (DINOv3)	96.4	98.0	97.3	94.0	91.8	91.9	98.0	93.8
	SubspaceAD$_{672}$ (DINOv3)	96.2	97.9	97.3	94.2	92.8	92.9	98.0	94.7
2-shot	SubspaceAD$_{448}$ (DINOv3)	97.1	98.3	97.6	94.5	92.9	92.7	98.2	94.3
	SubspaceAD$_{672}$ (DINOv3)	96.8	98.2	97.5	94.7	93.7	93.6	98.1	95.1
4-shot	SubspaceAD$_{448}$ (DINOv3)	97.6	98.6	97.8	94.8	93.7	93.4	98.3	94.7
	SubspaceAD$_{672}$ (DINOv3)	97.4	98.5	97.8	95.0	94.6	94.5	98.3	95.3

Table 8. Comparison of anomaly detection and localization performance on MVTec-AD and VisA in the **full-shot** setting. Best results are in **bold**, and second-best are underlined.

		MVTec-AD				VisA			
		Image-level		Pixel-level		Image-level		Pixel-level	
Setup	**Method**	AUROC	AUPR	AUROC	PRO	AUROC	AUPR	AUROC	PRO
Full-shot	AnomalyDINO	**99.5**	**99.8**	**98.2**	95.0	97.6	98.0	98.8	96.1
Full-shot	*SubspaceAD* (Ours)	99.2	99.6	**98.2**	**95.6**	**98.2**	**98.3**	**99.1**	**96.9**

Table 9. Hardware-normalized scoring head latency (H100). We measure the time to process features *after* extraction. SubspaceAD latency is invariant to K, whereas memory-bank retrieval scales with the number of shots.

Method	**Setting (K)**	**Scoring Time (ms / img)**
AnomalyDINO	1-shot	80.1
AnomalyDINO	2-shot	80.3
AnomalyDINO	4-shot	80.5
SubspaceAD (Ours)	1–4 shot	**74.1**

are challenging because their normal samples exhibit significant variation. This makes it difficult to form a single, compact subspace of normality from only a few shots. Furthermore, benign background artifacts (e.g., shadows, debris) that are not present in the few-shot support set may be incorrectly flagged as anomalies, since the model has no mechanism to infer that such regions belong to the normal background.

Outlook. Despite these limitations, the overall results demonstrate that even a simple, training-free subspace model surpasses far more complex approaches in both detection accuracy and efficiency. The clarity of its statistical formulation, combined with its strong few-shot generalization, highlights the potential of foundation-model representations when paired with lightweight, interpretable modeling. Future work may extend this direction by incorporating geometric or semantic priors to better handle logical and structural anomalies.

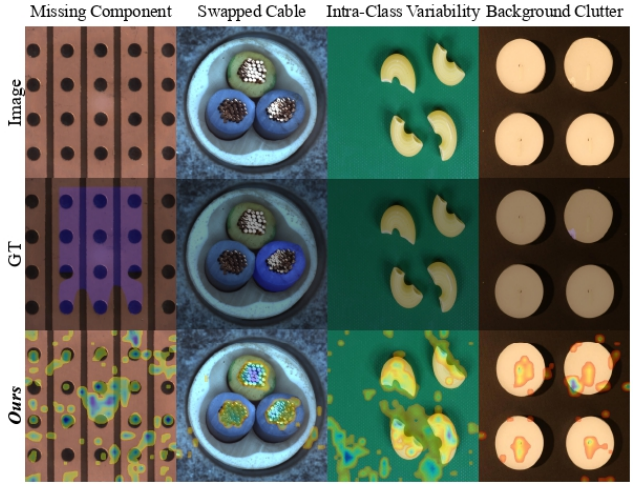


Figure 8. Qualitative failure cases. Each row shows the image, ground-truth mask, and our anomaly map. Examples include missed structural defects (e.g., missing transistor component), incorrect detection of cable swaps, and false positives from intra-class variability or background clutter.

G. Impact of Rotation-Agnostic Preprocessing

In the standard MVTec-AD dataset, the orientation of the *Transistor* object is strictly fixed, meaning misrotation explicitly constitutes an anomaly. To assess the effect of a uniform, rotation-agnostic preprocessing pipeline across all categories, we conducted an ablation where random rotations were applied to the normal support samples.

However, for orientation-dependent categories like *Transistor*,

Table 10. Inference time comparison with AnomalyDINO on MVTec-AD (1-shot, **448px**). Note that AnomalyDINO results are from the authors' paper [11], measured on an NVIDIA A40, while our measurements are obtained on an NVIDIA H100.

Method	Backbone	Params (M)	GPU	Time (ms / image)
WinCLIP [18]	CLIP ViT-B/16+	150.0	NVIDIA T4	389
AnomalyDINO [11]	DINOv2 ViT-S	21.7	NVIDIA A40	60
AnomalyDINO [11]	DINOv2 ViT-B	85.8	NVIDIA A40	84
AnomalyDINO [11]	DINOv2 ViT-L	303.3	NVIDIA A40	141
SubspaceAD (Ours) – DINOv2 Backbones (NVIDIA H100)				
SubspaceAD (Ours)	DINOv2 ViT-S	21.7	NVIDIA H100	36
SubspaceAD (Ours)	DINOv2 ViT-B	85.8	NVIDIA H100	56
SubspaceAD (Ours)	DINOv2 ViT-L	303.3	NVIDIA H100	112
SubspaceAD (Ours)	DINOv2 ViT-G	1100.0	NVIDIA H100	127
SubspaceAD (Ours) – DINOv3 Backbones (NVIDIA H100)				
SubspaceAD (Ours)	DINOv3 ViT-S	21.0	NVIDIA H100	16
SubspaceAD (Ours)	DINOv3 ViT-B	86.0	NVIDIA H100	21
SubspaceAD (Ours)	DINOv3 ViT-7B	7000.0	NVIDIA H100	330

rotating the normal samples during PCA fitting fundamentally alters the model's definition of normality. By incorporating rotated features into the normal subspace, the model inadvertently learns to accept rotational defects as normal variations, causing it to miss actual misrotation anomalies during inference.

Table 11 outlines the performance under this rotation-agnostic protocol. As expected, forcing the subspace to account for rotational variance leads to explicit performance drops compared to the standard aligned baseline (cf. Table 5). In the 1-shot setting, Image AUROC decreases by 7.5 percentage points, from 96.6% to 89.1%, while Pixel PRO drops by 6.3 percentage points, from 68.9% to 62.6%. This degradation persists in the 4-shot setting, where Pixel PRO decreases by 8.3 percentage points, from 72.7% to 64.4%. These results demonstrate that applying uniform rotational preprocessing is counterproductive for categories where orientation is a defining characteristic of the normal state.

Table 11. Few-shot anomaly segmentation results of *SubspaceAD* on the MVTec-AD *Transistor* category using a **rotation-agnostic** protocol. We report mean Image AUROC (%), Image AUPR (%), Pixel AUROC (%), and Pixel PRO (%) $\pm$ standard deviation across 5 seeds.

Shots	Img AUROC	Img AUPR	Pxl AUROC	Pxl PRO
1-shot	89.1 ± 7.5	84.7 ± 5.7	82.3 ± 3.1	62.6 ± 2.0
2-shot	92.6 ± 3.6	88.5 ± 4.7	83.6 ± 0.9	63.7 ± 0.9
4-shot	93.4 ± 4.0	90.3 ± 4.3	84.0 ± 1.9	64.4 ± 1.3