AnomalyDINO — Supplementary Material

A. Detailed experimental results

Further anomaly maps, predicted by AnomalyDINO, are presented in Figures 4 to 7. The full results per category for MVTec-AD and VisA are given in Tables 6 and 7, respectively.

The results presented in Tables 6 and 7 show that a) anomalies in some categories are more difficult to than others, b), that the performance of AnomalyDINO increases across the board with more available reference samples. In particular, more complex objects like 'PCB3' and 'PCB4' in VisA or 'Transistor' in MVTec-AD seem to benefit the most from more available reference samples. In addition, we see that also the variance in the reported metrics decreases with the number of nominal samples.

In this context, we observe a peculiarity of the few-shot regime (which does not occur for sufficiently populated $\mathcal{M}$). The results crucially depend on the chosen reference image(s).[6] The high variances, predominantly in the 1- and 2-shot setting, for category 'Capsule' for MVTec-AD, or 'Cashew' or 'PCB4' for VisA demonstrate this. We discuss the potential failure cases of choosing a sub-optimal reference sample in the following section (Appendix B.2).

Finally, we also report the full-shot performance, see Table 5. As the diversity within $\mathcal{M}$ is sufficiently high given the large number of reference samples, we only apply masing here (no augmentations). The results demonstrate that the performance further improves for all considered metrics. Notably, AnomalyDINO-S (672) achieves new state-of-the-art segmentation performance measured in (AU)PRO in the full-shot setting (see here, accessed 11/26/2024).

Table 5. **Full-shot results** on MVTec-AD and VisA with AnomalyDINO-S in the default setting (no std reported as results are deterministic when all samples are considered).

Dataset	Resolution	Detection			Segmentation		
		AUROC	F1-max	AP	AUROC	F1-max	PRO
MVTec-AD	448	99.3	98.8	99.7	97.9	61.8	93.9
	672	99.5	99.0	99.8	98.2	64.3	95.0
VisA	448	97.2	93.7	97.6	98.7	50.5	95.0
	672	97.6	94.5	98.0	98.8	53.8	96.1

B. Limitations and failure cases

The proposed method relies on similarities to patch representations captured in $\mathcal{M}$. Therefore, we can only expect the model to detect those anomalies caused by regions in the test images that are particularly different from patches

[6]Note that this holds for all one- and few-shot methods, not only for AnomalyDINO.

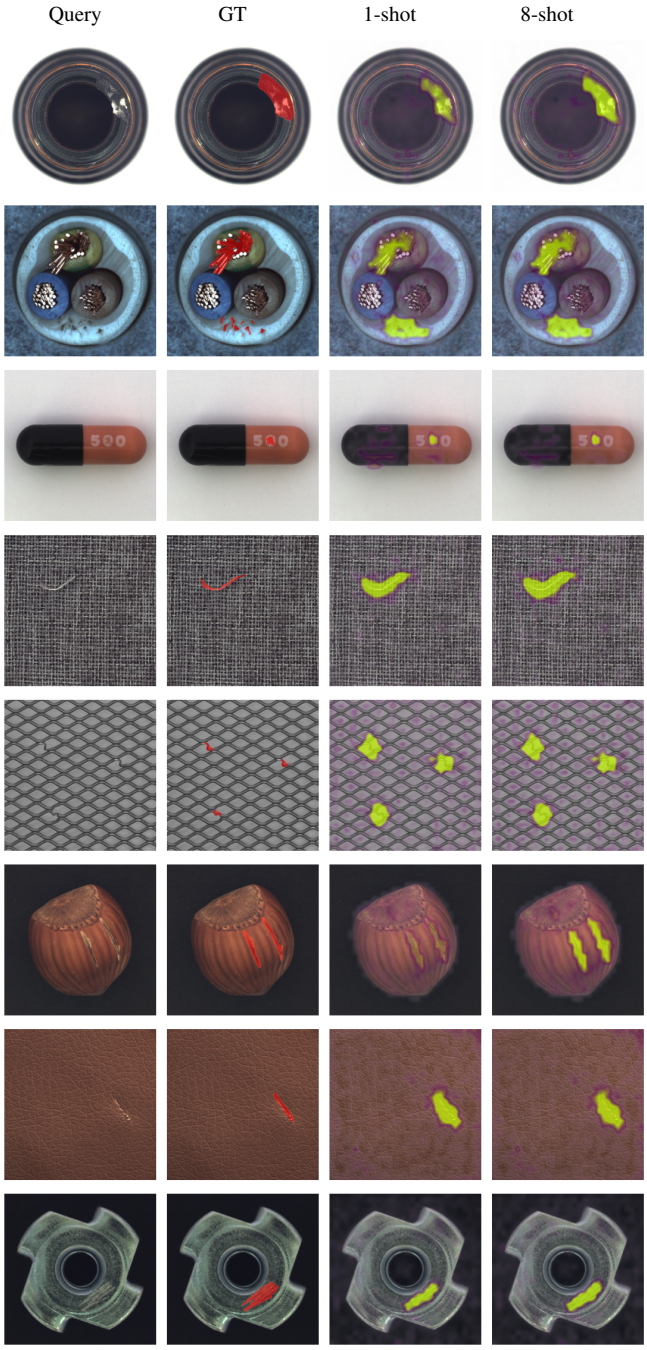


Figure 4. **Examples – MVTec-AD (1/2)**. Depicted are, from left to right, a test sample per category (Query), the ground truth anomaly annotation (GT), and the predicted anomaly map from AnomalyDINO-S (448) in the 1- and 8-shot settings. The color coding is normalized by the max. score over 'good' test samples.

Table 6. **Detailed results on MVTec-AD**. Reported are results on anomaly detection (image-level AUROC) and segmentation (PRO) of AnomalyDINO-S (672) with default preprocessing (mean and standard deviation over three independent runs, all results in %).

Shots	1-shot		2-shot		4-shot		8-shot		16-shot	
	AUROC	PRO	AUROC	PRO	AUROC	PRO	AUROC	PRO	AUROC	PRO
Bottle	99.7±0.2	95.9±0.3	99.9±0.1	96.3±0.0	99.9±0.1	96.7±0.1	100.0±0.0	96.5±0.4	99.9±0.1	96.4±0.4
Cable	92.7±0.8	89.4±0.4	92.4±1.1	89.5±0.4	93.8±0.9	90.4±0.3	95.2±0.3	90.5±0.2	95.1±0.6	90.5±0.6
Capsule	90.2±5.5	97.1±0.3	89.2±7.9	97.3±0.6	95.8±0.5	97.9±0.1	95.6±0.5	97.9±0.1	95.5±0.6	98.1±0.1
Carpet	100.0±0.0	97.8±0.0	100.0±0.0	97.9±0.0	100.0±0.0	97.8±0.0	100.0±0.0	97.8±0.0	100.0±0.0	97.8±0.0
Grid	99.1±0.2	97.2±0.1	99.2±0.4	97.2±0.1	99.5±0.3	97.2±0.1	99.5±0.1	97.2±0.0	99.7±0.3	97.0±0.2
Hazelnut	97.5±2.6	97.4±0.4	99.6±0.5	98.0±0.3	99.8±0.1	98.0±0.1	100.0±0.0	98.1±0.1	100.0±0.0	98.1±0.1
Leather	100.0±0.0	97.9±0.1	100.0±0.0	97.8±0.0	100.0±0.0	97.6±0.1	100.0±0.0	97.6±0.1	100.0±0.0	97.3±0.2
Metal nut	99.9±0.1	94.2±0.0	100.0±0.0	94.6±0.2	100.0±0.0	95.3±0.1	100.0±0.0	95.4±0.3	100.0±0.0	95.7±0.1
Pill	93.7±0.9	97.3±0.1	95.4±0.7	97.5±0.1	96.0±0.2	97.6±0.1	97.2±0.2	97.7±0.1	97.9±0.1	97.8±0.1
Screw	93.2±0.3	93.4±0.4	93.5±0.8	94.3±0.4	92.7±2.3	94.5±0.9	93.5±1.1	95.2±0.5	94.7±0.9	95.9±0.3
Tile	100.0±0.0	88.0±0.2	100.0±0.0	87.6±0.4	100.0±0.0	87.1±0.4	100.0±0.0	86.7±0.2	100.0±0.0	86.0±0.5
Toothbrush	97.4±0.5	94.0±0.8	98.1±1.0	94.7±0.3	97.5±0.6	94.7±0.3	97.7±1.6	95.1±0.8	98.1±1.8	95.8±1.0
Transistor	90.9±1.2	67.3±2.1	89.4±4.6	68.4±3.1	93.2±2.2	70.6±1.4	96.2±1.3	75.3±1.9	97.6±0.3	78.2±1.4
Wood	98.0±0.2	94.7±0.1	98.0±0.1	94.6±0.0	97.9±0.2	94.6±0.1	98.3±0.4	94.4±0.3	98.3±0.6	94.2±0.4
Zipper	97.4±0.9	89.2±1.2	98.9±0.4	90.2±0.4	99.0±0.4	91.2±0.3	99.6±0.1	91.1±0.4	99.6±0.3	91.7±0.5
Mean	96.6±0.4	92.7±0.1	96.9±0.7	93.1±0.2	97.7±0.2	93.4±0.1	98.2±0.2	93.8±0.1	98.4±0.1	94.0±0.1

Table 7. **Detailed results on VisA**. Reported are results on anomaly detection (image-level AUROC) and segmentation (PRO) of AnomalyDINO-S (672) with default preprocessing (mean and standard deviation over three independent runs, all results in %).

Shots	1-shot		2-shot		4-shot		8-shot		16-shot	
	AUROC	PRO	AUROC	PRO	AUROC	PRO	AUROC	PRO	AUROC	PRO
Candle	87.9±0.3	96.8±0.4	89.4±3.0	97.0±0.2	91.3±2.9	97.2±0.1	93.5±1.2	97.3±0.2	94.5±0.5	97.6±0.2
Capsules	98.4±0.5	95.1±0.7	98.9±0.1	95.5±0.2	99.2±0.1	96.3±0.4	99.2±0.1	96.7±0.2	99.2±0.2	97.2±0.3
Cashew	86.1±3.6	96.1±0.9	89.4±3.8	96.7±0.7	94.5±0.7	97.4±0.5	95.3±0.6	97.3±0.2	96.0±0.3	97.3±0.1
Chewinggum	98.0±0.4	92.0±1.0	98.6±0.4	92.9±0.3	98.8±0.2	93.0±0.1	98.8±0.2	93.1±0.3	98.8±0.2	93.0±0.3
Fryum	94.8±0.5	93.2±0.2	96.5±0.2	93.9±0.3	97.0±0.1	94.5±0.4	97.6±0.4	94.9±0.3	97.9±0.2	95.1±0.0
Macaroni1	87.5±1.1	97.5±0.3	87.5±0.9	97.9±0.3	89.5±1.4	98.3±0.2	90.1±1.7	98.6±0.2	90.4±1.1	98.7±0.1
Macaroni2	62.2±4.3	92.0±0.7	66.9±1.9	93.0±0.4	70.0±1.7	93.9±0.8	74.9±0.4	95.0±0.6	77.6±0.8	95.7±0.3
PCB1	91.5±2.0	92.6±0.2	91.2±2.7	92.5±0.5	94.0±2.1	93.3±0.5	95.5±0.5	93.9±0.2	96.8±0.7	94.2±0.2
PCB2	84.8±1.2	89.9±0.2	88.1±2.5	90.7±0.3	91.1±1.7	91.4±0.2	92.6±0.3	92.0±0.1	93.2±0.1	92.5±0.2
PCB3	84.9±3.3	88.5±1.3	89.4±3.8	90.8±0.5	94.3±0.4	91.7±0.4	95.6±0.2	93.1±0.3	96.5±0.3	93.9±0.3
PCB4	79.9±13.7	78.5±6.8	87.4±11.3	82.0±6.1	96.2±2.6	84.1±1.5	98.0±0.3	87.9±2.3	99.0±0.4	90.4±1.8
Pipe fryum	92.7±2.7	98.0±0.0	93.3±1.2	97.9±0.2	94.6±1.9	97.8±0.1	95.0±1.6	97.6±0.1	97.2±0.9	97.7±0.1
Mean	87.4±1.2	92.5±0.5	89.7±1.3	93.4±0.6	92.6±0.9	94.1±0.1	93.8±0.3	94.8±0.2	94.8±0.2	95.3±0.2

of the given reference sample(s). Our experiments reveal that this may lead to some specific failure cases.

B.1. Semantic Anomalies

The first relates to the distinction between low-level sensory anomalies and high-level semantic anomalies. Semantic anomalies might be present because a logical rule or a specific semantic constraint is violated. In contrast, AnomalyDINO is developed with low-level sensory anomalies in mind. Consider, for instance, the anomalies shown in Figure 8.

While the cable with anomaly type 'Bent Wire' (Fig-

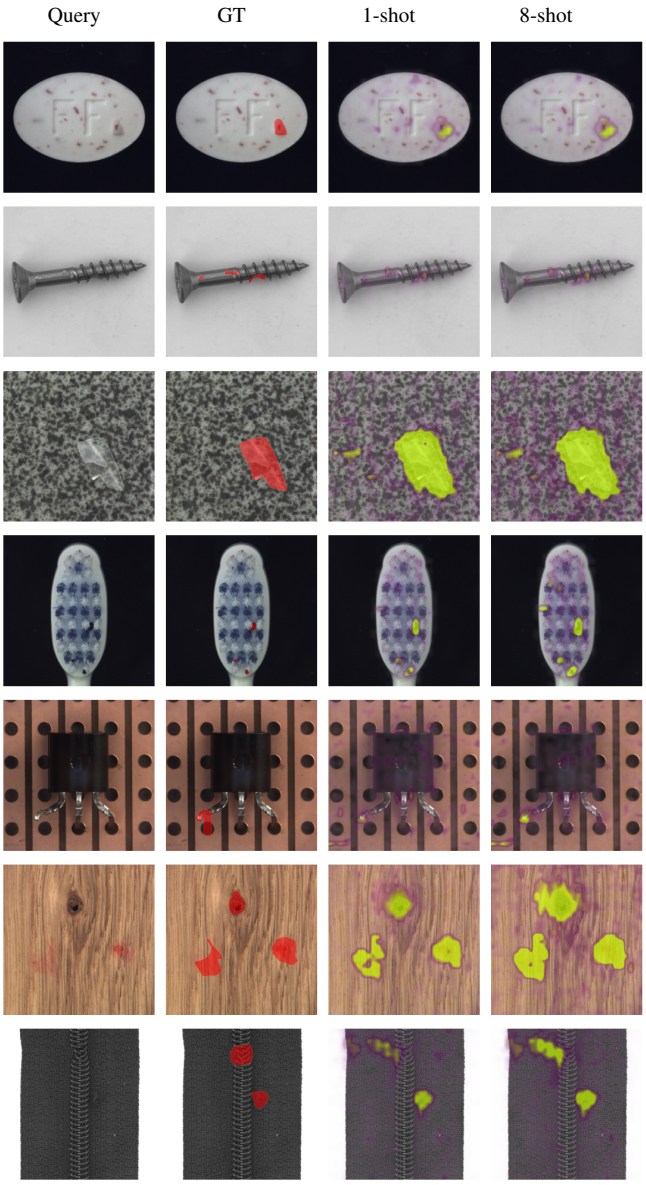


Figure 5. **Examples – MVTec-AD (2/2)**. See Fig. 4 for a description.

ure 8b) contains patches that cannot be well matched to patches of the reference image, this does not apply to the anomaly 'Cable Swap' (Figure 8c). The latter shows a *semantic anomaly* with two blue wires, while a nominal image of a 'Cable' should depict all three different cable types. As a result, all test patches of Figure 8b can be matched well with reference patches in $\mathcal{M}$, and thus, AnomalyDINO does not detect this anomaly type. Evaluating the detection performance for this specific failure case, 'Cable Swap', our proposed method essentially performs on chance level, giving a detection AUROC of 50.2% ($\pm$ 4.9%).

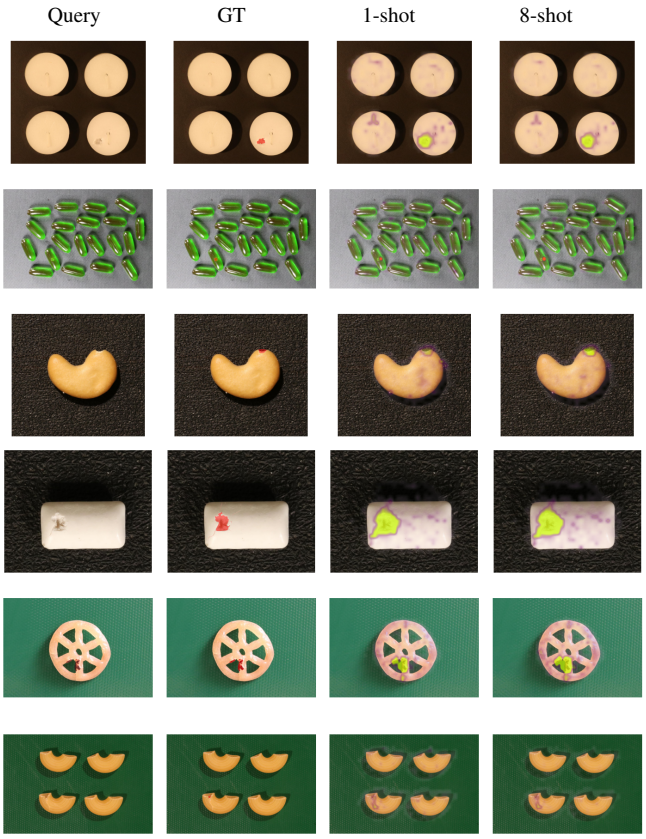


Figure 6. **Examples – VisA (1/2)**. Depicted are, from left to right, a test sample per category (Query), the ground truth anomaly annotation (GT), and the predicted anomaly map from AnomalyDINO-S (448) in the 1- and 8-shot settings. The color bar is normalized by the maximum score on the 'good' test samples (per category). For the category 'Capsules' (left column, second from top) we changed the color map for better visibility. Best viewed at a higher zoom level as some anomalies are quite small.

B.2. The importance of informative reference samples

The second failure case occurs if the reference sample(s) does not resemble all concepts of normality, therefore $\mathcal{M}$ does not capture all variations of the nominal distribution p_{norm}. As an illustration, consider the nominal samples of 'Capsule' in MVTec-AD, which may be rotated in such a way that the imprinted text is hidden (see Figure 9b). Therefore, parts of the text of a nominal test sample may be falsely recognized as anomalies. Due to the strong dependency on a suitable reference sample, we observe higher AD variances for some products in the one-shot setting, as shown in Tables 6 and 7. We like to remark, that this is only relevant to the few-shot setting, and with an increasing number of reference samples (and higher diversity of nominal patches in $\mathcal{M}$), the variance decreases notably.

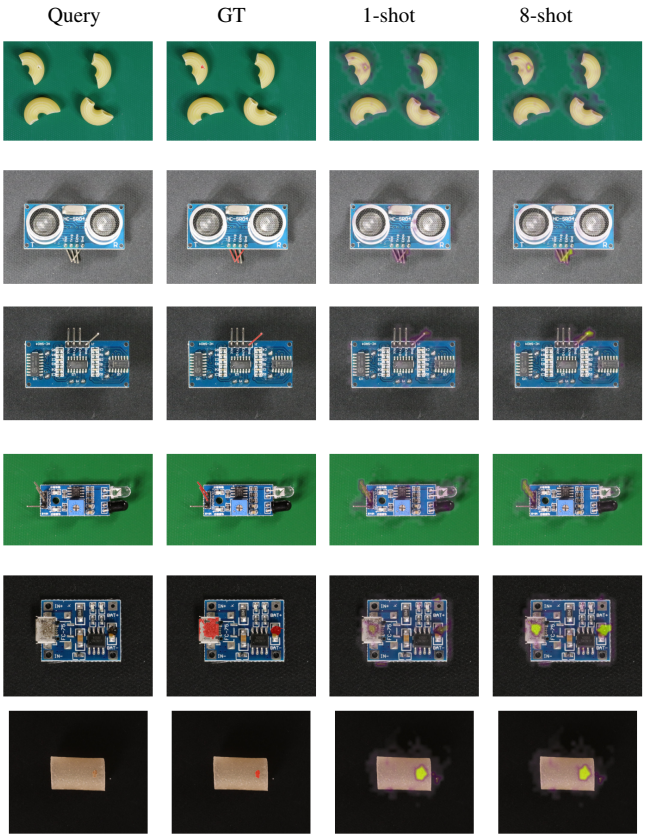
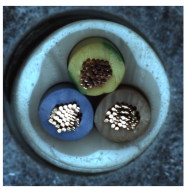
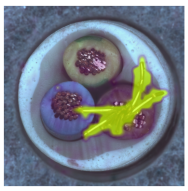


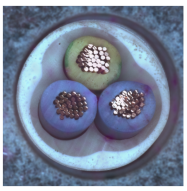
Figure 7. **Examples – VisA (2/2)**. See Fig. 6 for the description.



(a) Reference sample (No anomalies)



(b) Anomaly type 'Bent Wire'



(c) Anomaly type 'Cable Swap'

Figure 8. **Example of a semantic anomaly** in category 'Cable' (MVTec-AD). Depicted are a nominal reference sample, a sensory anomaly (Figure 8b), and a semantic anomaly (Figure 8c) with anomaly maps predicted by AnomalyDINO (1-shot).

(a) Reference sample (left) and estimated anomaly map of 'good' test sample (right).

(b) Reference sample (left) and estimated anomaly map of 'good' test sample (right).

Figure 9. **Example of an uninformative reference sample** in category 'Capsule' (MVTec-AD). Some reference samples do not resemble the full concept of normality (here, the sample in Figure 9b does not show the text on the capsule, i.e., any nominal sample with text be considered anomalous). Anomaly maps predicted by AnomalyDINO based on the depicted reference sample.

C. Ablation Study

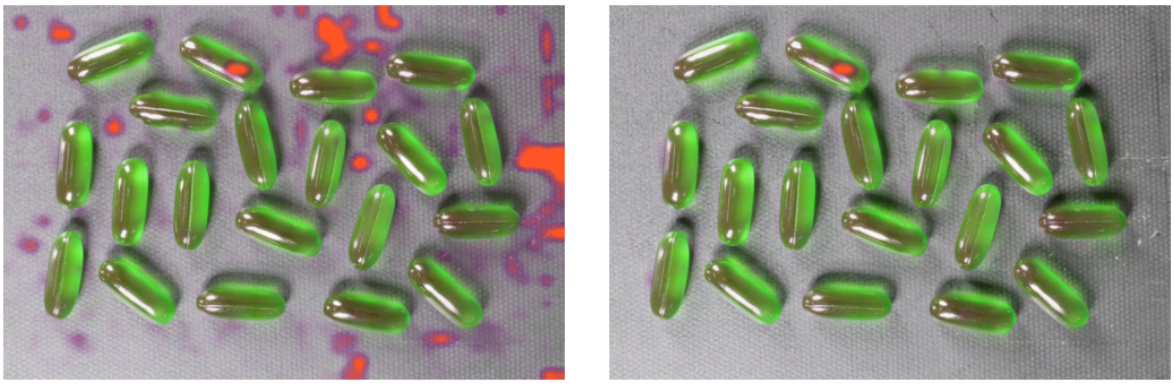
C.1. Preprocessing

As discussed in Section 3, we consider two potential preprocessing steps in our pipeline: masking and rotations. We mask out irrelevant background patches, whenever the zero-shot segmentation of DINOv2 captures the object correctly (see Figure 2). Discarding background patches helps to mitigate the problem of potential background noise, thereby reducing the number of false positives. A representative example, here from the category 'Capsules' from the VisA dataset, is depicted in Figure 10. Without masking, the method would correctly predict the depicted sample to show an anomaly, but for the wrong reason (high anomaly scores caused by background noise/contamination). In contrast, the irrelevant background areas are discarded by our masking approach, such that the anomalous region is correctly identified. We conclude that masking is essential to faithfully detect anomalies whenever higher variations in the background are expected—and as the issue of low variations is particularly pressing in the few-shot regime, caused by the minimal amount of nominal reference samples, suitable masking can significantly boost the performance in these cases. This is showcased by the superior anomaly localization performance A quantitative analysis is provided in the next paragraphs.

In addition to masking (which decreases the size of $\mathcal{M}$), we consider augmenting the reference sample with rotations (which increases the size and diversity of $\mathcal{M}$). Here we distinguish the 'agnostic' scenario, where we do not know a priori about potential rotations, and augment by default. Whenever we know about potential rotations of reference or test samples, we can deactivate the augmentation to reduce the size of the memory bank $\mathcal{M}$, the time to construct $\mathcal{M}$, as well as the test time itself.

The preprocessing decision per category for MVTec-AD

(a) Test sample (left), the same test sample with ground-truth anomaly annotation (center), and a magnified view of a region with strong background artifacts (right).



(b) Anomaly map predicted by AnomalyDINO (1-shot) without masking (left) and with masking (right).

Figure 10. **Visualization of the effect of masking** in the presence of high background noise for the category 'Capsules' in VisA (1-shot). As in Figure 6 we depict the anomaly map for 'Capsules' using a different colormap (red instead of yellow) for better visibility. Best viewed on a higher zoom level.

and VisA, inferred (solely) from the first nominal reference sample in X_{ref} (to comply with the one-shot setting), are given in Table 8.

Table 8. **Default preprocessing steps for MVTec-AD and VisA**. We do not mask textures, as indicated by (T). In addition, we do not apply masking when the masking test on the first train sample failed, as indicated by (MT). See Section 3 and Figure 2 for further discussion and visualization.

MVTec-AD	Mask?	Rotation?	
Object		informed	agnostic
Bottle	✗(MT)	✗	✓
Cable	✗(MT)	✗	✓
Capsule	✓	✗	✓
Carpet	✗(T)	✗	✓
Grid	✗(T)	✗	✓
Hazelnut	✓	✓	✓
Leather	✗(T)	✗	✓
Metal nut	✗(MT)	✗	✓
Pill	✓	✗	✓
Screw	✓	✓	✓
Tile	✗(T)	✗	✓
Toothbrush	✓	✗	✓
Transistor	✗(MT)	✗	✓
Wood	✗(T)	✗	✓
Zipper	✗(MT)	✗	✓

VisA	Mask?	Rotation?	
Object		informed	agnostic
Candle	✓	✗	✓
Capsules	✓	✗	✓
Cashew	✓	✗	✓
Chewinggum	✓	✗	✓
Fryum	✓	✗	✓
Macaroni1	✓	✗	✓
Macaroni2	✓	✗	✓
PCB1	✓	✗	✓
PCB2	✓	✗	✓
PCB3	✓	✗	✓
PCB4	✓	✗	✓
Pipe fryum	✓	✗	✓

Effect of preprocessing on detection performance As discussed in Section 3 (and in the previous paragraph), suitable means to fill $\mathcal{M}$ and preprocess test samples, influence the detection performance. The results per object for MVTec-AD and VisA are given in Figures 11 and 12, respectively.

Rotation Increasing the diversity of nominal patch representations in $\mathcal{M}$ by rotating the reference sample can significantly improve the detection performance. Consider, e.g., the category 'Screw' in MVTec-AD, where the detection AUROC can be boosted from 65.6% to 89.2% in the 1-shot setting. This is intuitive, as the test samples of 'Screw' are taken from various angles. With sufficiently many reference samples, such data augmentation is not necessary anymore, but for the few-shot setting, we see major improvements.

The same holds—although to a lesser extent—for the categories 'Hazelnut', 'Cable', and 'Wood' (all MVTec-AD). However, we also observe that rotations of the reference sample can also *decrease* the detection performance in some categories, namely 'PCB1/2/3' and 'Macaroni1' (all VisA), and to a small extent also 'Transistor' (MVTec-AD).

We attribute this to the fact the specific anomalies for

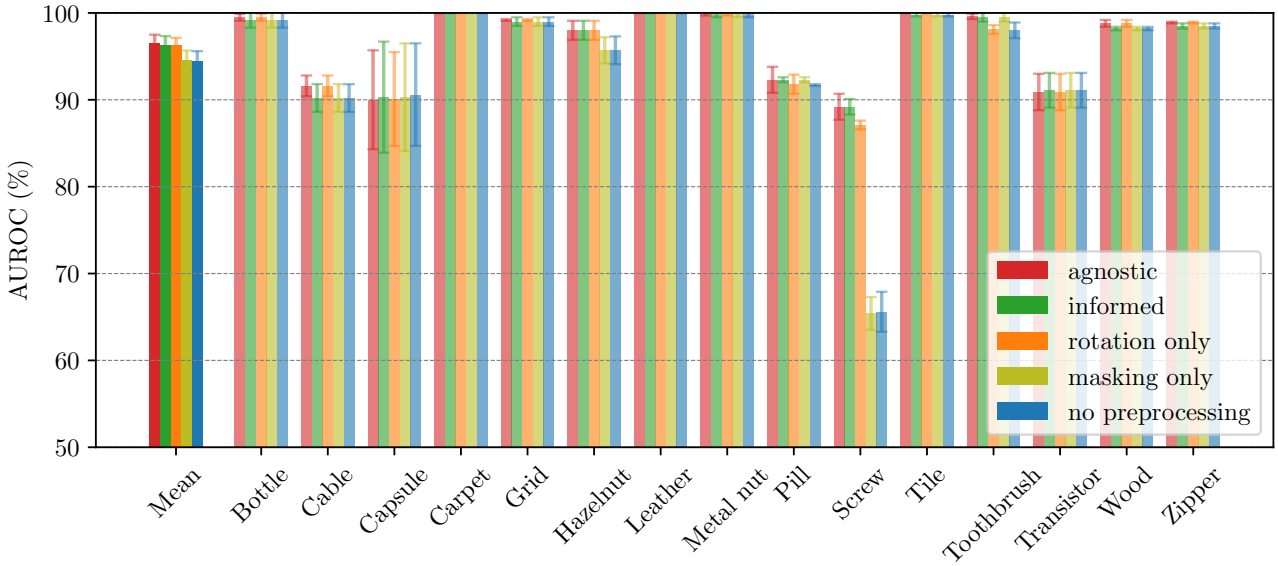


Figure 11. **Effect of preprocessing on MVTec-AD**. Anomaly detection of AnomalyDINO-S (448) in the 1-shot setting for different choices of the preprocessing pipeline (detection AUROC on image-level in %, mean and standard deviation over three independent runs).

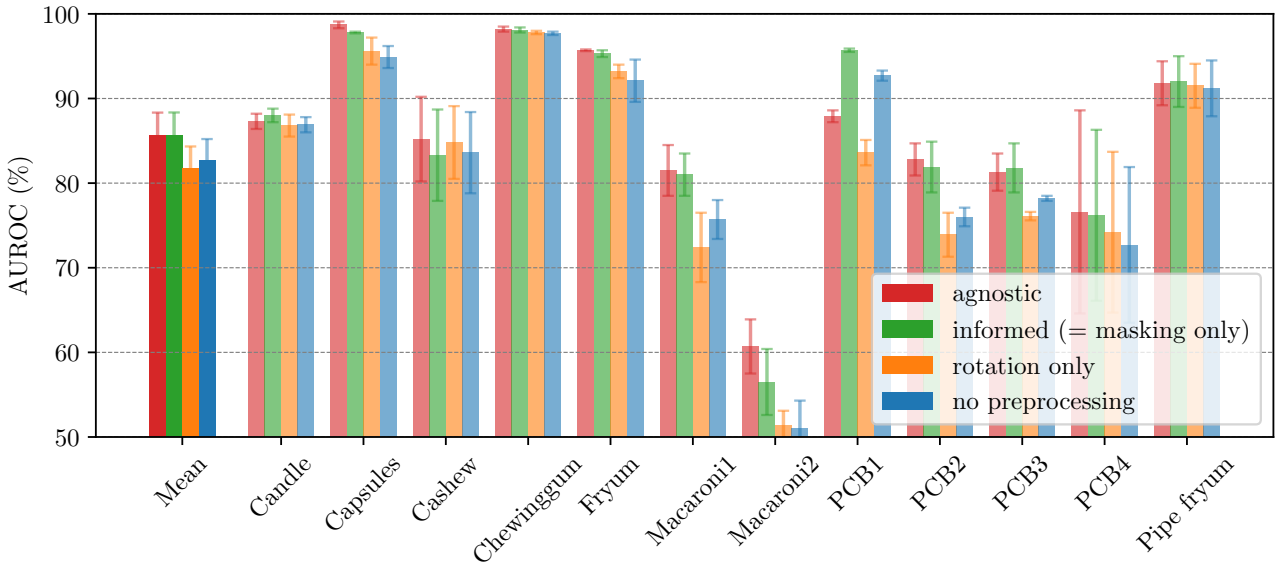


Figure 12. **Effect of preprocessing on VisA.** Anomaly detection with AnomalyDINO-S (448) in the 1-shot setting for different choices of the preprocessing pipeline (detection AUROC on image-level in %, mean and standard deviation over three independent runs). For VisA, the 'informed' scenario is equivalent to only applying masking (all categories), while 'agnostic' is equivalent to masking and augmentations (all categories), see Table 8.

these printed circuit boards contain rotated or bent connectors (see the PCB examples in Figure 6, right column). And rotations of nominal samples may falsely reduce the distance between those patches depicting bent connectors and the nominal memory bank $\mathcal{M}$.

Such a failure case based on a suboptimal preprocessing decision is depicted in Figure 13. The anomaly refers to a rotated transistor (anomaly type 'misplaced'), and when (falsely) rotating the reference sample, such anomalies will not be detected—in contrast to the 'informed' preprocessing (right side of Figure 13). This highlights the importance of carefully designing the pre- and postprocessing pipeline for each object/product considered.

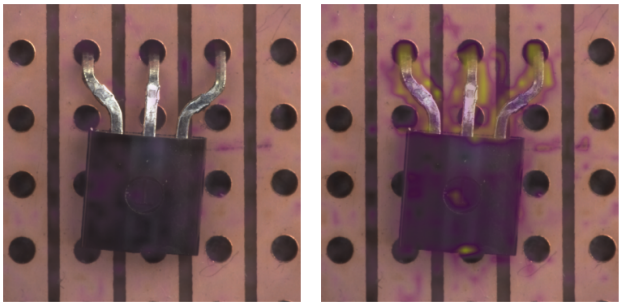


Figure 13. **Rotations as anomalies** ('misplaced' transistor from MVTec-AD). The left anomaly map is estimated from a reference sample *with* rotations ('agnostic'), and the anomaly is not detected. In contrast, the right anomaly map is based on a reference sample *without* rotations ('informed'), and the anomaly is successfully detected. This example highlights the importance of a carefully designed preprocessing pipeline for each object.

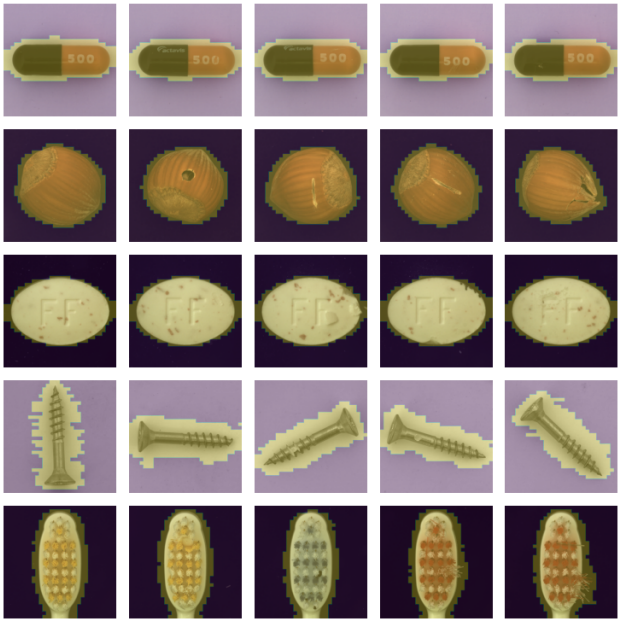


Figure 14. **Masking examples from MVTec-AD** for all categories that passed the masking test (on a single train sample, see Table 8).

Masking We utilize the zero-shot masking capabilities of DINOv2 to keep the overhead for this preprocessing step minimal. By applying a threshold to the first principal component [30], we can typically distinguish between the background and the foreground.

We also employ a straightforward rule-based strategy to enhance the robustness and generalizability of the PAC-based masking in industrial settings: ensuring that the center crop of the image is predominantly occupied by the object of interest. This minor adjustment is necessary because

Figure 15. **Masking examples from VisA**.

industrial images often differ significantly from the data on which DINOv2 was originally trained. In addition, we use dilation and morphological closing to improve the quality of the mask. In our experiments, we applied masking only to the test samples as the size of $\mathcal{M}$ did not matter in the few-shot regime. Across the board, we see that the proposed masking technique improves the detection performance—in many categories even significantly, e.g., for 'Capsules', 'Macaroni1/2', or 'PCB1/2/3/4' (all VisA). We leave further improvements and the exploration of more advanced masking techniques for future work.

Runtime analysis The design choices in our pipeline influence the run time of the proposed method. To see the potential effects we measure the inference time per sample on MVTec-AD in various scenarios together with the time to set up the memory bank $\mathcal{M}$. The runtime was assessed utilizing a single NVIDIA A40 GPU, consistently employed throughout all experiments detailed in this paper (each ex-

Table 9. **Runtime analysis** on MVTec-AD (mean and std of inference time over all 1725 test samples, and mean and std of time to populate the memory bank for each object) for different shots (Tab. 9a), preprocessing choices (Tab. 9b), sample resolutions (Tab. 9c) and model sizes (Tab. 9d). All times are reported in seconds, measured on a single NVIDIA A40 with GPU warmup and CUDA kernel synchronization. The default setting is 1-shot, agnostic preprocessing, model size S, and resolution 448 (underlined for reference).

(a) Runtime in dependence of shots.

Shots	Inference	Memory Bank
1	$0.060_{\pm 0.012}$	$0.52_{\pm 0.04}$
2	$0.059_{\pm 0.012}$	$0.85_{\pm 0.07}$
4	$0.059_{\pm 0.012}$	$1.58_{\pm 0.08}$
8	$0.060_{\pm 0.011}$	$3.05_{\pm 0.17}$
16	$0.063_{\pm 0.012}$	$6.02_{\pm 0.39}$
full (masking only)	$0.067_{\pm 0.010}$	$16.62_{\pm 4.43}$
full (agnostic)	$0.130_{\pm 0.044}$	$93.72_{\pm 20.56}$

(b) Runtime in dependence of preprocessing choices.

Mask?	Rotate?	Inference	Memory Bank
no	no	$0.055_{\pm 0.011}$	$0.17_{\pm 0.03}$
no	yes	$0.055_{\pm 0.012}$	$0.51_{\pm 0.05}$
yes	no	$0.068_{\pm 0.012}$	$0.18_{\pm 0.03}$
yes	yes	$0.067_{\pm 0.012}$	$0.51_{\pm 0.04}$
informed		$0.059_{\pm 0.016}$	$0.22_{\pm 0.11}$
agnostic		$0.060_{\pm 0.012}$	$0.52_{\pm 0.04}$

(c) Runtime in dependence of image resolution.

Resolution	Inference	Memory Bank
224	$0.043_{\pm 0.010}$	$0.31_{\pm 0.04}$
448	$0.060_{\pm 0.012}$	$0.52_{\pm 0.04}$
672	$0.086_{\pm 0.014}$	$0.75_{\pm 0.06}$
896	$0.141_{\pm 0.021}$	$1.26_{\pm 0.05}$

(d) Runtime in dependence of model size.

Model Size	Inference	Memory Bank
S (21 M)	$0.060_{\pm 0.012}$	$0.52_{\pm 0.04}$
B (86 M)	$0.084_{\pm 0.021}$	$0.77_{\pm 0.05}$
L (300 M)	$0.141_{\pm 0.029}$	$1.24_{\pm 0.06}$
G (1,100 M)	$0.306_{\pm 0.034}$	$2.67_{\pm 0.14}$

periment can be executed on a single NVIDIA A40 GPU). The runtime is measured with GPU warmup and CUDA kernel synchronization for a fair comparison.

The results are given in Table 9. The average inference time per sample with AnomalyDINO-S (448) amounts to approximately 60ms in the few-shot regime ($\approx$ 16.7 samples/s), and only moderately increases with a larger memory bank (to 67ms ($+11\%$) for the full-shot scenario without augmentations). Compared to SOTA competitors in the one-shot regime, this is at least one order of magnitude faster (see Figure 3). Without any preprocessing steps, the 1-shot inference time amounts to approximately 55ms per sample ($\approx$ 18 samples/s). When applying both masking and rotations, the runtime increases from 55ms to 67ms, a moderate increase of roughly 23% (compared to the scenario without any preprocessing steps). The informed scenario can reduce the time to build the memory bank, but inference time is only affected for larger sample sizes. As expected, higher resolutions and larger architectures lead to increased runtimes.

C.2. Scoring

In Section 3 we investigate different ways of aggregating the anomaly scores $\mathcal{D}$ on patch-level (in our case, distances to the nominal memory bank $\mathcal{M}$) to an image-level anomaly score via a statistic q. Note that the segmentation results are therefore not affected by the choice of q. A standard choice is upsampling the patch distances of lower resolution to the full resolution of the test image using bilinear interpolation, then applying Gaussian smoothing operation (we follow [34] and set $\sigma = 4.0$), to obtain an anomaly map $\mathcal{A}$, and set $q = \max(\mathcal{A})$. We also evaluate two potential alternatives of q, our default choice $q = \text{mean}(H_{0.01}(\mathcal{D}))$ (mean of the 1% highest entries in $\mathcal{D}$) and $q = \max(\mathcal{D})$.

The results in Table 10 show that just the maximum patch distance leads to already good results while upsampling and smoothing the patch distances giving slightly weaker results (maximum of $\mathcal{A}$).

Taking the mean of all patch distances above the 99% quantile ($q = \text{mean}(H_{0.01}(\mathcal{D}))$) improves above the standard choice. We did not optimize over the percentile and instead fixed it to 99%. Typically, the number of patches per image ranges between 200 and 1000 (depending on resolution and masking), such that 2 between 10 patches are considered. For specific products/objects and expected anomaly types, other statistics q might be (more) suitable.

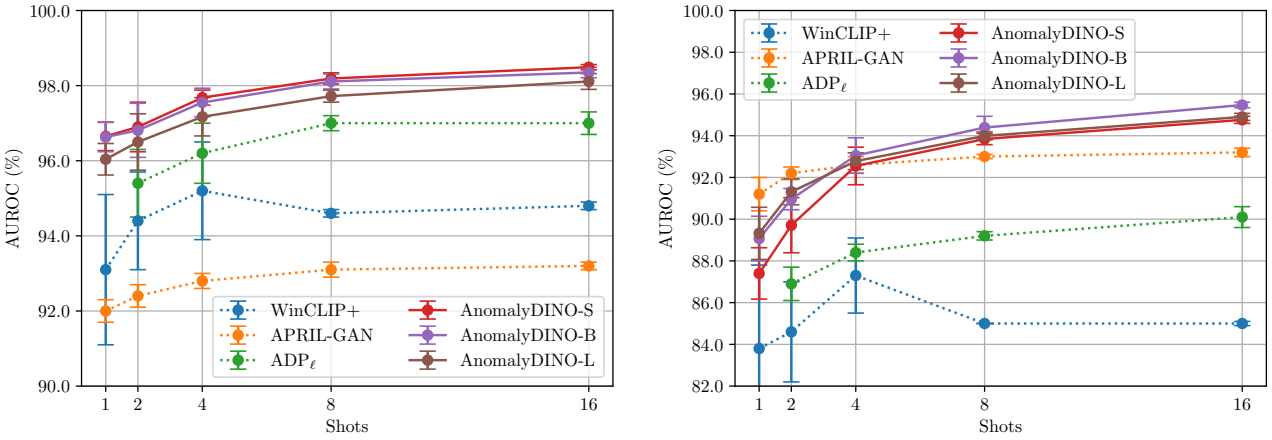
C.3. Architecture Size

DINOv2 is available in different distillation sizes (S, B, L, and G). This section analyzes the implications of choosing different backbone sizes for AnomalyDINO. The comparison including the best competing methods is depicted in Figure 16. We see that different architecture sizes have indeed an effect on the image-level AUROC.

On MVTec-AD, we see that AnomalyDINO-S performs best, followed by the next larger model AnomalyDINO-B

Table 10. **Effect of aggregation statistics** q on the detection performance on MVTec-AD and VisA, evaluated for AnomalyDINO-S (448). $\mathcal{D}$ denotes the set of patch distances to the nominal memory bank, $H_{0.01}(\mathcal{D})$ the 1% highest entries thereof, and $\mathcal{A}$ denotes the anomaly map derived from $\mathcal{D}$ (anomaly scores for each pixel, after upsampling and smoothing, see Section 3). All results in %.

	Scoring	$q = \text{mean}(H_{0.01}(\mathcal{D}))$			$q = \max(\mathcal{D})$			$q = \max(\mathcal{A})$		
	Shots	AUROC	F1-max	AP	AUROC	F1-max	AP	AUROC	F1-max	AP
MVTec-AD	1	$96.5_{\pm 0.4}$	$96.0_{\pm 0.2}$	$98.1_{\pm 0.3}$	$95.0_{\pm 0.5}$	$94.7_{\pm 0.2}$	$97.4_{\pm 0.4}$	$94.9_{\pm 0.7}$	$94.6_{\pm 0.6}$	$97.5_{\pm 0.4}$
	2	$96.7_{\pm 0.8}$	$96.5_{\pm 0.4}$	$98.1_{\pm 0.7}$	$95.5_{\pm 1.0}$	$95.3_{\pm 0.5}$	$97.4_{\pm 0.6}$	$95.0_{\pm 1.3}$	$94.8_{\pm 0.8}$	$97.4_{\pm 0.9}$
	4	$97.6_{\pm 0.1}$	$97.0_{\pm 0.3}$	$98.4_{\pm 0.3}$	$96.6_{\pm 0.2}$	$96.0_{\pm 0.1}$	$97.8_{\pm 0.4}$	$96.3_{\pm 0.4}$	$95.7_{\pm 0.2}$	$98.1_{\pm 0.3}$
	8	$98.0_{\pm 0.1}$	$97.4_{\pm 0.1}$	$99.0_{\pm 0.2}$	$97.2_{\pm 0.1}$	$96.5_{\pm 0.4}$	$98.6_{\pm 0.1}$	$97.0_{\pm 0.0}$	$96.3_{\pm 0.1}$	$98.6_{\pm 0.1}$
	16	$98.3_{\pm 0.1}$	$97.7_{\pm 0.2}$	$99.3_{\pm 0.0}$	$97.6_{\pm 0.2}$	$96.9_{\pm 0.3}$	$98.9_{\pm 0.1}$	$97.4_{\pm 0.1}$	$96.6_{\pm 0.2}$	$98.8_{\pm 0.1}$
VisA	1	$85.6_{\pm 1.5}$	$83.1_{\pm 1.1}$	$86.6_{\pm 1.3}$	$82.4_{\pm 1.9}$	$81.4_{\pm 1.0}$	$84.0_{\pm 1.7}$	$80.5_{\pm 1.3}$	$80.6_{\pm 0.8}$	$82.1_{\pm 0.9}$
	2	$88.3_{\pm 1.8}$	$84.8_{\pm 1.2}$	$89.2_{\pm 1.3}$	$85.1_{\pm 1.8}$	$82.5_{\pm 1.3}$	$86.4_{\pm 1.1}$	$83.3_{\pm 1.8}$	$82.2_{\pm 1.1}$	$84.5_{\pm 1.5}$
	4	$91.3_{\pm 0.8}$	$87.5_{\pm 1.0}$	$91.8_{\pm 0.7}$	$88.4_{\pm 0.3}$	$84.9_{\pm 0.6}$	$89.0_{\pm 0.4}$	$86.8_{\pm 1.4}$	$84.3_{\pm 0.9}$	$87.6_{\pm 1.3}$
	8	$92.6_{\pm 0.1}$	$88.6_{\pm 0.2}$	$92.9_{\pm 0.2}$	$90.0_{\pm 0.3}$	$86.3_{\pm 0.2}$	$90.6_{\pm 0.5}$	$88.8_{\pm 0.5}$	$85.3_{\pm 0.3}$	$89.7_{\pm 0.3}$
	16	$93.8_{\pm 0.1}$	$89.9_{\pm 0.3}$	$94.2_{\pm 0.3}$	$91.6_{\pm 0.5}$	$87.2_{\pm 0.7}$	$92.2_{\pm 0.4}$	$90.5_{\pm 0.3}$	$86.8_{\pm 0.5}$	$91.5_{\pm 0.4}$



(a) Detection AUROC for MVTec-AD.

(b) Detection AUROC for VisA.

Figure 16. **Effect of model size** on detection AUROC for MVTec-AD and VisA (mean and std over three seeds). The image resolution of AnomalyDINO is set to 672. Note, that the results for WinCLIP+ for 8 and 16 shots are those of the WinCLIP re-implementation [21].

(which might contrast the common belief that larger models always perform better). In particular, all architecture sizes outperform the closest competitor (ADP$_\ell$).

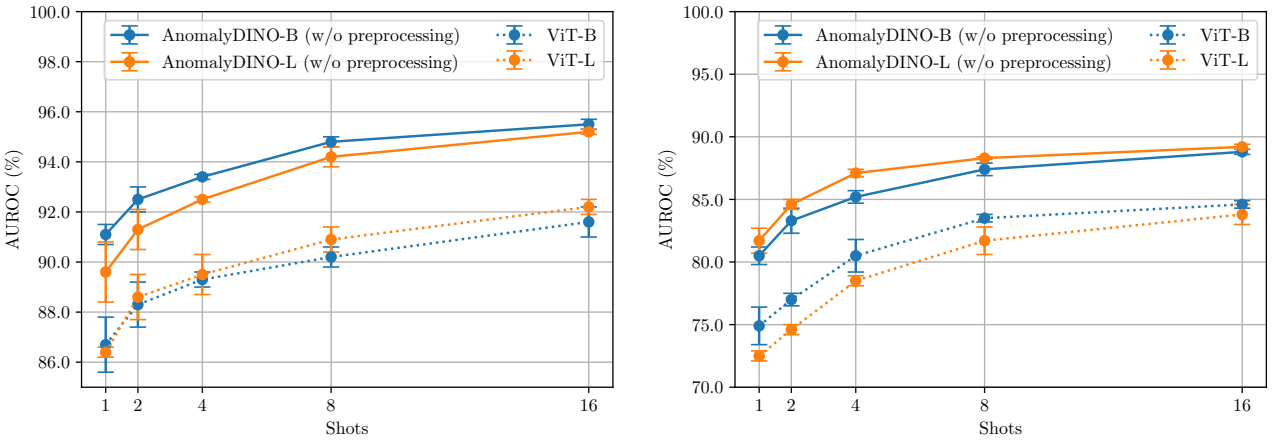
Regarding VisA, larger architectures slightly outperform our default setting, AnomalyDINO-S. All architecture sizes are on par with APRIL-GAN at $k = 4$, but outperform all competitors for $k > 4$. Figure 16 also showcases that the performance of the proposed method scales preferably with the number of reference samples.

C.4. Backbone Choice

In our experiments, we find that DINOv2 provides excellently suited features for few-shot AD. This is already evident in Fig. 3, here we investigate the effect on the detection performance on MVTec-AD and VisA in more detail.

We compare the performance in few-shot AD of ViT-B and ViT-L pre-trained on ImageNet to that of AnomalyDINO-B and AnomalyDINO-L (utilizing DINOv2 in distillation sizes B and L). To ensure comparability, we deactivate any pre- and postprocessing for AnomalyDINO (no augmentations, no zero-shot masking) and set the image resolution to 224 for all models.

The results are depicted in Figure 17. We observe that DINOv2 significantly improves the detection performance compared to those of the ViT pre-trained on the image classification tasks, giving a performance gain of at least $+4\%$ AUROC on MVTec-AD and VisA. This demonstrates that the features extracted by DINOv2 are better suited com-



(a) Detection AUROC for MVTec-AD.

(b) Detection AUROC for VisA.

Figure 17. **Effect of backbone choice** DINOv2 (trained in a self-supervised fashion) compared to ViT (supervised training on ImageNet) on detection AUROC for MVTec-AD and VisA (mean and std over three seeds). For better comparability, image resolution of AnomalyDINO is 224 for all models.

pared to those of ImageNet-pretrained ViT-{B/L}.

In Sections 3 and 4 we demonstrate that further (substantial) improvements are possible with suitable pre- and postprocessing like augmentations and zero-shot masking (which is not possible based on the features from supervised features) and that the image resolution (or the effective patch size) can also greatly boost performance. See also Fig. 16 for the performance in dependence of the model size.

D. Extending AnomalyDINO to the Batched Zero-Shot Setting

We extend the proposed method to the *batched* zero-shot setting. Recall, that in this setting all test samples X_{test} are provided (or at least a sufficiently large batch), but *no* (labeled) training or reference samples. The underlying (and necessary) assumption to meaningfully predict anomalies solely based on test samples, is that the majority of samples (or in our case, patches) at test time are from the nominal data distribution (see e.g., [22]).

We need to alter our method, outlined in Section 3, only slightly to adapt it to the batched setting. We score a test sample $\mathbf{x}^{(j)} \in X_{\text{test}}$ in comparison to all remaining test sample $X_{\text{test}} \setminus \{\mathbf{x}^{(j)}\}$, following the idea of mutual scoring [24]. For each test sample $\mathbf{x}^{(j)} \in X_{\text{test}}$ we therefore collect all patch representations not belonging to $\mathbf{x}^{(j)}$ in a memory bank, again utilizing DINOv2 as patch-level feature extractor f,

$$\mathcal{M}_j := \bigcup_{\mathbf{x}^{(i)} \in X_{\text{test}} \setminus \{\mathbf{x}^{(j)}\}} \left\{ \mathbf{p}_m \mid f(\mathbf{x}^{(i)}) = (\mathbf{p}_1, \ldots, \mathbf{p}_n),\, m \in [n] \right\} . \tag{5}$$

We need to infer anomaly scores for each patch representation $\mathbf{p}_{\text{test}}$ of $\mathbf{x}^{(j)}$. We could again assess the distances between $\mathbf{p}_{\text{test}}$ and $\mathcal{M}_j$ based on the distance to the nearest neighbor, as done in Equation (2). Note, however, that $\mathcal{M}_j$ may now contain nominal *and* anomalous patches. Thus, the nearest neighbor approach is not suitable anymore: the nearest neighbor in $\mathcal{M}_j$ for (the representation of) an anomalous patch might also be abnormal, and the resulting distance therefore not informative.

A simple solution, based on the assumption that the majority of patches are nominal, is to replace the nearest neighbor with a suitable aggregation statistic over the distribution of patches distances in $\mathcal{M}_j$

$$\mathcal{D}(\mathbf{p}_{\text{test}}, \mathcal{M}_j) := \left\{ d(\mathbf{p}_{\text{test}}, \mathbf{p}) \mid \mathbf{p} \in \mathcal{M}_j \right\} , \tag{6}$$

where we again use the cosine distance d, defined in Equation (3). Specifically, we can again make use of the tail value at risk—now for the lowest quantile as we are interested in the tail behavior of the distribution of distances to the nearest neighbors—to derive a patch-level anomaly score

$$s(\mathbf{p}_{\text{test}}) := \text{mean}\big(L_\alpha\big(\mathcal{D}(\mathbf{p}_{\text{test}}, \mathcal{M}_j)\big)\big) , \tag{7}$$

where $L_\alpha(\mathcal{D})$ contains the values below the α quantile of $\mathcal{D}$. We set $\alpha = 0.1\%$ as anomalous patches are rare by assumption, and $\mathcal{D}(\mathbf{p}_{\text{test}}, \mathcal{M}_j)$ large enough to accurately estimate the tail of the distribution.[7] The image-level score $s(\mathbf{x}_{\text{test}})$ is again given by aggregating the patch-level anomaly score following Equation (4). Computation of the

[7] For MVTec-AD the total number of test patches extracted by DINOv2 (ViT-S) at a resolution of 448 range between 43.008 and 171.008 per category, and for VisA between 163.200 and 334.464.

cosine distances and the proposed aggregation statistics can be effectively implemented as matrix operations on GPU such that the batched zero-shot inference time for AnomalyDINO amounts to roughly 60 ms/sample for MVTec-AD at a resolution of 448 (again measured on an NVIDIA A40 GPU). Some resulting anomaly maps in the batched zero-shot setting are depicted in Figure 18.

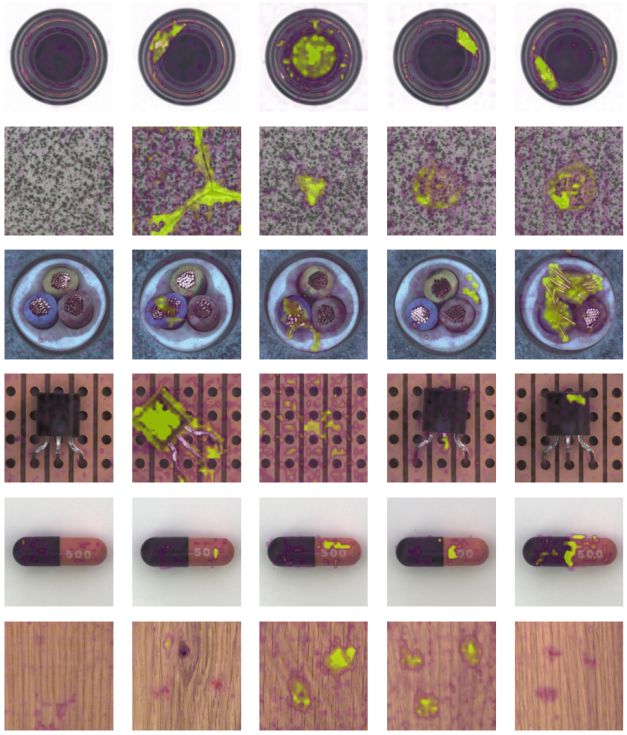


Figure 18. **Anomaly maps for the batched zero-shot setting** on MVTec-AD. The left-most sample in each category is a 'good' test sample for reference, followed by four randomly picked samples with anomalies.

E. Broader Impacts

Advancing few-shot visual anomaly detection methods can offer various benefits by enhancing manufacturing quality control through the rapid identification of defects with minimal nominal examples, which improves efficiency, reduces waste, and improves overall safety in the product lifecycle. Similar positive benefits can be expected outside the industrial domain, e.g., for healthcare diagnostics or environmental monitoring. It is, however, essential to recognize the shortcomings of automated anomaly detection systems. We believe that simpler methods can be adapted more quickly, monitored more effectively, and are therefore more reliable. In this context, awareness of the risks of over-reliance must be heightened (see Appendix B for identified failure cases of the proposed method). In addition, strong visual anomaly detectors could also lead to potentially malicious or unintended uses. To address these concerns, including potential privacy infringements and possible socioeconomic impacts of automation, strategies such as establishing robust data governance, and implementing strict privacy protections are essential. Additionally, investing in workforce development can help manage the socioeconomic effects of automation and leverage the full potential of strong visual anomaly detection.