Supplementary:
Towards Total Recall in Industrial Anomaly Detection

A. Implementation Details

We implemented our models in Python 3.7 [51] and PyTorch [37]. Experiments are run on Nvidia Tesla V4 GPUs. We used torchvision ImageNet-pretrained models from torchvision and the PyTorch Image Models repository [53]. By default, following [10] and [14], *PatchCore* uses a WideResNet50-backbone [57] for direct comparability. Patch-level features are taken from feature map aggregation of the final outputs in blocks 2 and 3. For all nearest neighbour retrieval and distance computations, we use `faiss` [27].

B. Full MVTec AD comparison

This section contains a more detailed comparison on MVTec AD. We include more models and a more finegrained performance comparison on all MVTec AD sub-datasets where available. In the main part of the paper this has been referenced in §4.2. The corresponding result tables are S1, S2 and S3. We observe that *PatchCore*-25% solves six of the 15 MVTec datasets and achieves highest AUROC performance on most datasets and in average.

Figure S3 show Precision-Recall and ROC curves for *PatchCore* variants as well as reimplemented, comparable methods SPADE [10] and PaDiM [14] using a WideResNet50 backbone. We also plot classification error both at 100% recall and under a F1-optimal threshold to give a comparable working point. As can be seen, *PatchCore* achieves consistently low classification errors with defined working points as well, with near-optimal Precision-Recall and ROC curves across datasets, in contrast to SPADE and PaDiM.

Finally, Table S4 showcases the detailed performance on all MVTec AD subdatasets for larger imagesizes (280×280) and a WideResNet-101 backbone for further performance boosts using PatchCore-1%, which allows for efficient anomaly detection at inference time even with larger images.

C. Additional Ablations & Details

C.1. Detailed Low-Shot experiments

This section offers detailed numerical values to the low-shot method study provided in the main part of this work (§4.5). The results are included in Table S5 and we find consistently higher numbers for detection and anomaly localization metrics.

C.2. Dependency on pretrained networks

We tested *PatchCore* with different backbones, the results are shown in S6. We find that results are mostly stable over the choice of different backbones. The choice of WideResNet50 was made to be comparable with SPADE and PaDiM.

C.3. Influence of image resolution

Next we study the influence of image size on performance. In the main paper we have used 224×224 to be comparable with prior work. In Figure S4 we vary the image size from 288×288, 360×360 to 448×448 and the neighborhood sizes (P) within 3, 5, 7, and 9. We observe slightly increased detection performance and the performance saturates for *PatchCore* . For anomaly segmentation we observe a consistent increase, so if good localization is of importance, this is an ingredient to validate over.

C.4. Remaining Misclassifications

The high image-level anomaly detection performance allows us to look into all remaining misclassifications in detail. We compute the working point (threshold above which scores are considered anomalous) using the F1-optimal point. With this threshold a total of 19 false-positive and 23 false-negative errors remain, all of which are visualized in Figures S1 and S2. Each segmentation map was normalized to the threshold value, so in some cases background scores are pronounced disproportionally.

Looking at Figure S1, we find that the majority of false-positive errors either stem from a) (in blue) ambiguity in labelling , i.e., image changes that could also be potentially labelled as anomalous, and b) (in orange) very high nominal variance,

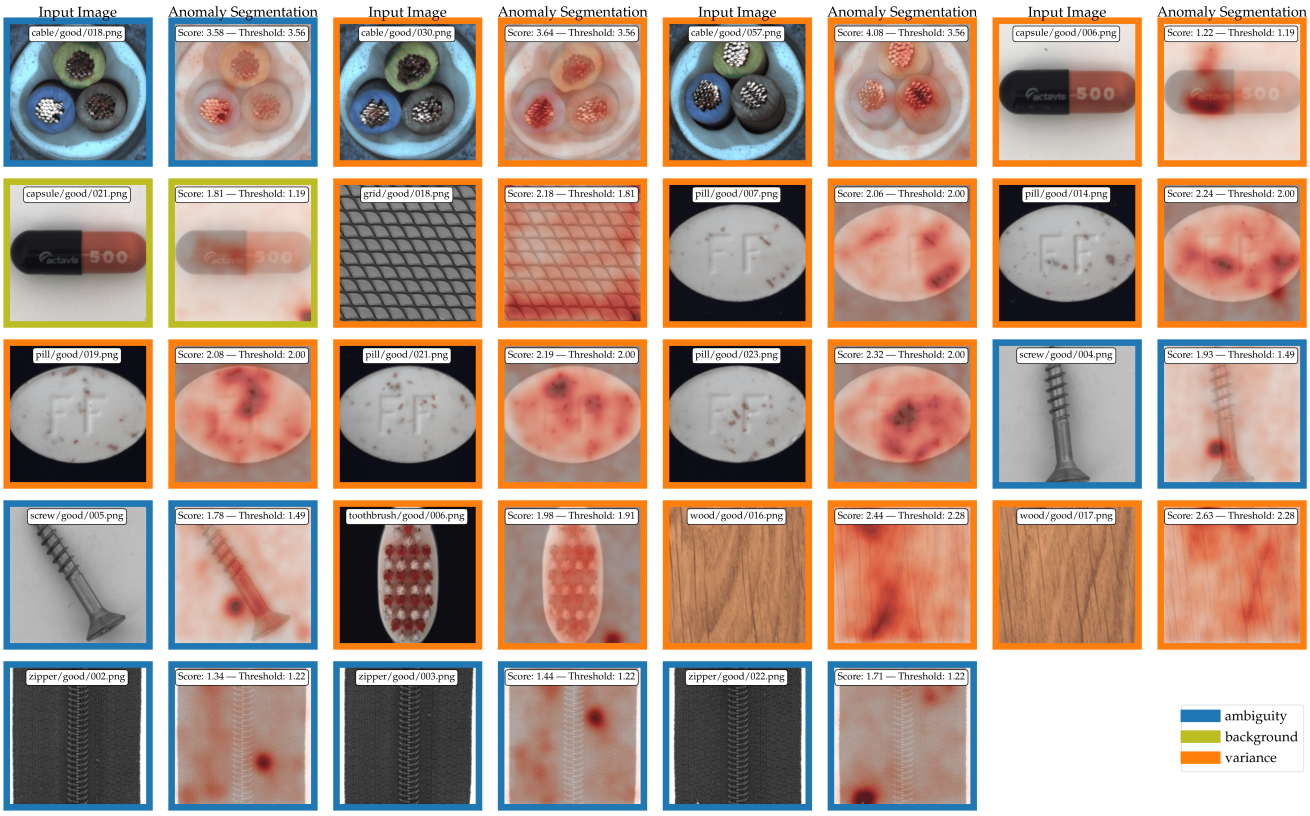

Figure S1. Visualization of remaining false positive classifications (under F1-optimal thresholding). Colors denote different error sources. Orange denotes high degrees of nominal variance mistaken for anomalies, blue denotes misclassifications due to anomalies in the labelling context and olive denotes variance in the background mistaken for anomalous content.

resembling potential anomalies . While the former can hardly be addressed by proposed methods, the latter could be addressed by offering some form of adaptation to the nominal data. However, as *PatchCore* outperforms adaptive methods, such adaptation would show most promise operating alongside pretraining-based methods such as *PatchCore* .

To understand false-negative errors made, we include in Figure S2 the generated segmentation maps and ground-truth masks. As can be seen, a large part of anomalies are localized well, however with insufficient weight placed on the anomalous regions, and could potentially be addressed by some means of postprocessing. Other misclassifications are caused mostly by either high degrees of nominal variance that gets mistaken for anomalous context, and finegrained anomalies that could be captured when moving to higher image resolutions. The amount of completely missed anomalies is small in comparison, and in one case caused by image preprocessing cropping out the actual anomalous region.

C.5. Local Awareness and Subsampling

For completeness we repeat the Figures 4 and 5 from the main paper with included PRO score results in S5 and S6.

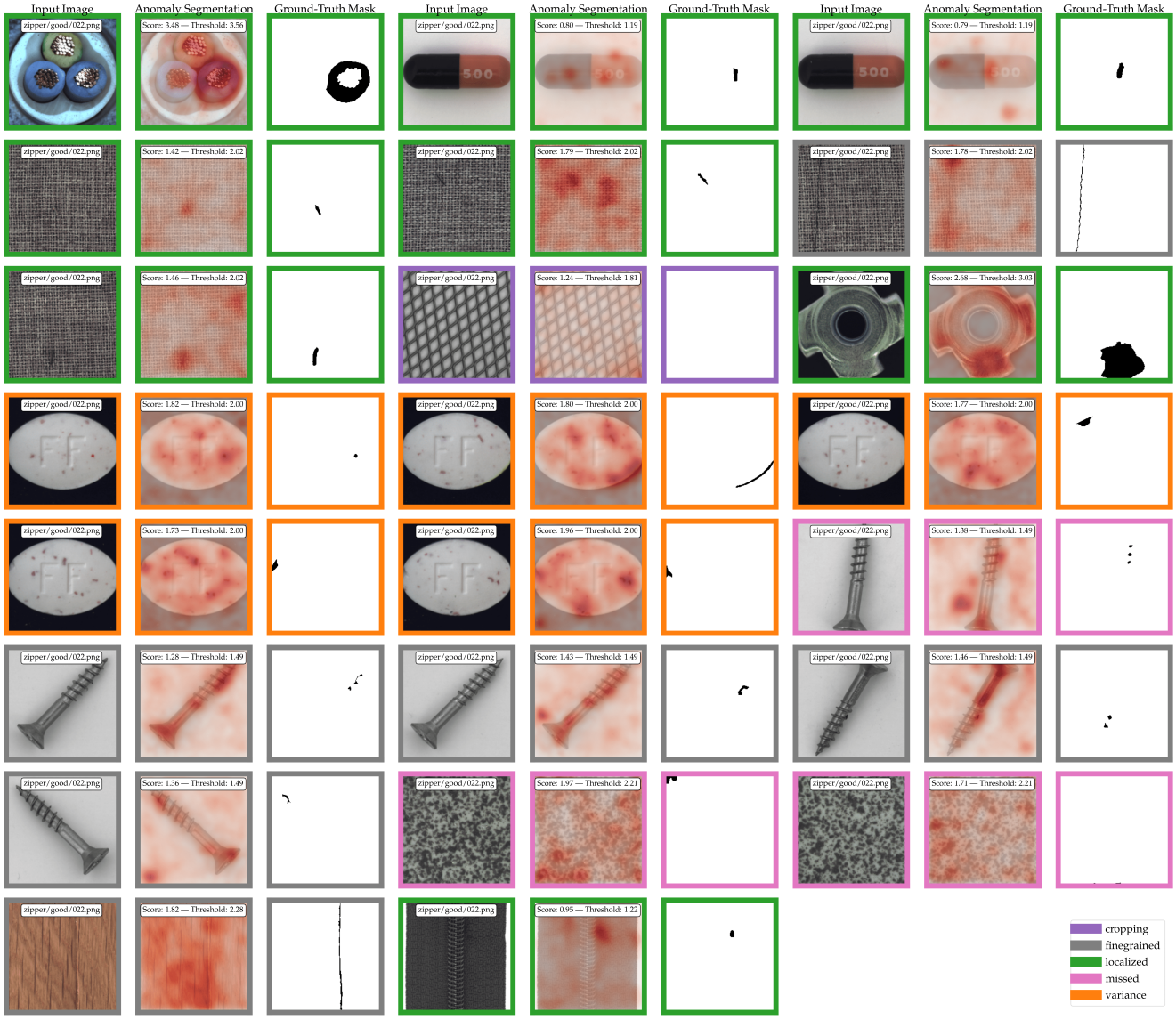

Figure S2. Visualization of remaining false-negative classifications (under F1-optimal thresholding). Colors denote different error sources. Orange denotes high degrees of nominal variance mistaken for anomalies, green denotes actually localized anomalies, but too little weight placed on these anomalies, pink stands for anomalies that were not recovered, purple denotes anomalies missed due to cropping-based image-processing (one anomaly in total), and gray stands for finegrained anomalies that could be recovered when operating on higher image resolutions.

Table S1. Anomaly Detection Performance (AUROC) on MVTec AD [5]. PaDiM* denotes a result from [14] with a backbone specifically selected for the task of image-level anomaly detection, which we could not reproduce.

↓ Method \Dataset →	**Avg**	Bottle	Cable	Capsule	Carpet	Grid	Hazeln.	Leather	Metal Nut	Pill	Screw	Tile	Toothb.	Trans.	Wood	Zipper
GeoTrans [20]	67.2	74.4	78.3	67.0	43.7	61.9	35.9	84.1	81.3	63.0	50.0	41.7	97.2	86.9	61.1	82.0
GANomaly [2]	76.2	89.2	75.7	73.2	69.9	70.8	78.5	84.2	70.0	74.3	74.6	79.4	65.3	79.2	83.4	74.5
DSEBM [58]	70.9	81.8	68.5	59.4	41.3	71.7	76.2	41.6	67.9	80.6	99.9	69.0	78.1	74.1	95.2	58.4
OCSVM [3]	71.9	99.0	80.3	54.4	62.7	41.0	91.1	88.0	61.1	72.9	74.7	87.6	61.9	56.7	95.3	51.7
ITAE [25]	83.9	94.1	83.2	68.1	70.6	88.3	85.5	86.2	66.7	78.6	**100**	73.5	**100**	84.3	92.3	87.6
SPADE [10]	85.5	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
CAVGA-R_u [52]	90	96	92	93	88	84	97	89	82	86	81	97	89	99	79	96
PatchSVDD [56]	92.1	98.6	90.3	76.7	92.9	94.6	92.0	90.9	94.0	86.1	81.3	97.8	**100**	91.5	96.5	97.9
DifferNet [42]	94.9	99.0	95.9	86.9	92.9	84.0	99.3	97.1	96.1	88.8	96.3	99.4	98.6	91.1	**99.8**	95.1
PaDiM [14]	95.3	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
MahalanobisAD [40]	95.8	**100**	95.0	95.1	**100**	89.7	99.1	**100**	94.7	88.7	85.2	**99.8**	96.9	95.5	99.6	97.9
PaDiM* [14]	97.9	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
PatchCore−25	**99.1**	**100**	**99.5**	**98.1**	**98.7**	98.2	**100**	**100**	**100**	96.6	98.1	98.7	**100**	**100**	99.2	99.4
PatchCore−10	99.0	100	99.4	97.8	98.7	97.9	**100**	**100**	**100**	96.0	97.0	98.9	99.7	**100**	99.0	**99.5**
PatchCore−1	99.0	100	99.3	98.0	98.0	**98.6**	**100**	**100**	99.7	**97.0**	96.4	99.4	**100**	99.9	99.2	99.2

Table S2. Anomaly Segmentation Performance on MVTec [5], as measured in pixelwise AUROC.

↓ Method \Dataset →	**Avg**	Bottle	Cable	Capsule	Carpet	Grid	Hazeln.	Leather	Metal Nut	Pill	Screw	Tile	Toothb.	Trans.	Wood	Zipper
vis. expl. VAE [31]	86	87	90	74	78	73	98	95	94	83	97	80	94	93	77	78
AE_{SSIM} [5]	87	93	82	94	87	94	97	78	89	91	96	59	92	90	73	88
γ-VAE + grad. [15]	88.8	93.1	88.0	91.7	72.7	97.9	98.8	89.7	91.4	93.5	97.2	58.1	98.3	93.1	80.9	87.1
CAVGA-R_u [52]	89	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
PatchSVDD [56]	95.7	98.1	96.8	95.8	92.6	96.2	97.5	97.4	98.0	95.1	95.7	91.4	98.1	97.0	90.8	95.1
SPADE [10]	96.0	98.4	97.2	**99.0**	97.5	93.7	**99.1**	97.6	98.1	96.5	98.9	87.4	97.9	94.1	88.5	96.5
PaDiM [14]	97.5	98.3	96.7	98.5	**99.1**	97.3	98.2	99.2	97.2	95.7	98.5	94.1	**98.8**	**98.5**	94.9	98.5
PatchCore−25	**98.1**	**98.6**	98.4	98.8	99.0	**98.7**	98.7	**99.3**	**98.4**	97.4	**99.4**	95.6	98.7	96.3	95.0	98.8
PatchCore−10	**98.1**	**98.6**	**98.5**	98.9	**99.1**	**98.7**	98.7	**99.3**	**98.4**	**97.6**	**99.4**	95.9	98.7	96.4	**95.1**	**98.9**
PatchCore−1	98.0	98.5	98.2	98.8	98.9	98.6	98.6	**99.3**	**98.4**	97.1	99.2	**96.1**	98.5	94.9	**95.1**	98.8

Table S3. Anomaly Segmentation Performance on MVTec [5], as measured in PRO [%] [5, 10].

↓ Method \Dataset →	**Avg**	Bottle	Cable	Capsule	Carpet	Grid	Hazeln.	Leather	Metal Nut	Pill	Screw	Tile	Toothb.	Trans.	Wood	Zipper
AE_{SSIM} [5]	69.4	83.4	47.8	86.0	64.7	84.9	91.6	56.1	60.3	83.0	88.7	17.5	78.4	72.5	60.5	66.5
Student [6]	85.7	91.8	86.5	91.6	69.5	81.9	93.7	81.9	89.5	93.5	92.8	**91.2**	86.3	70.1	72.5	93.3
SPADE [10]	91.7	95.5	90.9	93.7	94.7	86.7	**95.4**	97.2	**94.4**	**94.6**	96.0	75.6	**93.5**	**87.4**	87.4	92.6
PaDiM [14]	92.1	94.8	88.8	93.5	96.2	94.6	92.6	97.8	85.6	92.7	94.4	86.0	93.1	84.5	**91.1**	95.9
PatchCore−25	93.4	**96.2**	92.5	**95.5**	**96.6**	96.0	93.8	**98.9**	91.4	93.2	**97.9**	87.3	91.5	83.7	89.4	**97.1**
PatchCore−10	**93.5**	96.1	**92.6**	**95.5**	**96.6**	95.9	93.9	**98.9**	91.3	94.1	**97.9**	87.4	91.4	83.5	89.6	**97.1**
PatchCore−1	93.1	95.9	91.6	**95.5**	96.5	**96.1**	93.8	**98.9**	91.2	92.9	97.1	88.3	90.2	81.2	89.5	97.0

Table S4. Anomaly Detection and Localization Performance (AUROC) on MVTec AD [5] with PatchCore−1 using larger images (280 × 280) and a WideResNet101 backbone.

↓ Metric \Dataset →	**Avg**	Bottle	Cable	Capsule	Carpet	Grid	Hazeln.	Leather	Metal Nut	Pill	Screw	Tile	Toothb.	Trans.	Wood	Zipper
PatchCore−1, Hierarchies (2, 3), Imagesize 280																
AUROC	99.4	100	99.6	98.2	98.4	99.8	100	100	100	97.2	98.9	98.9	100	100	99.5	99.9
pwAUROC	98.2	98.6	98.4	99.1	98.7	98.7	98.8	99.3	98.8	97.8	99.3	96.1	98.8	96.4	95.1	98.9
PRO	94.4	96.6	93.8	96.0	97.4	96.8	91.2	99.1	94.8	94.0	97.5	89.5	95.5	84.8	91.7	97.8
PatchCore−1, Hierarchies (1, 2, 3), Imagesize 280																
AUROC	99.2	100	99.7	98.1	98.2	98.3	100	100	100	97.1	99.0	98.9	98.9	99.7	99.9	99.7
pwAUROC	98.4	98.6	98.7	99.1	98.7	98.8	98.8	99.3	99.0	98.6	99.5	96.3	98.9	97.1	95.2	99.0
PRO	95.0	96.6	94.6	96.3	97.5	97.0	91.5	99.1	95.4	96.0	98.1	90.0	95.8	85.9	92.0	98.0

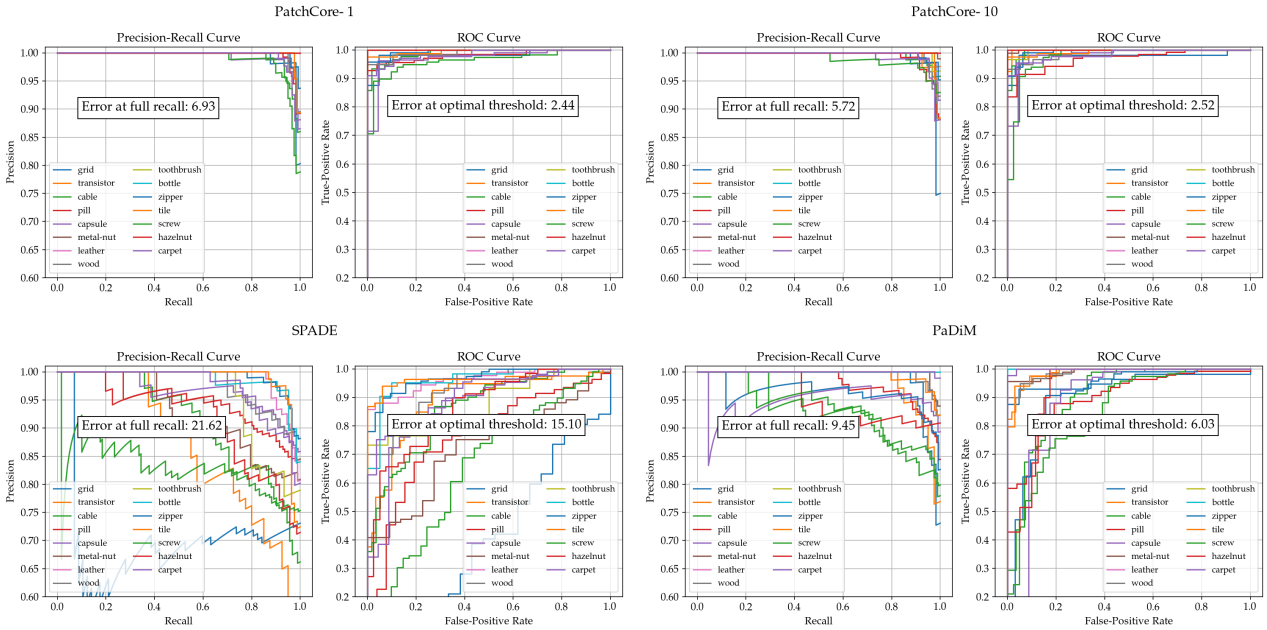

Figure S3. Precision-Recall curves (left) and ROC curves (right) for *PatchCore* , variants and comparable methods SPADE [10] and PaDiM [14]. Different colors in the lines correspond to difference MVTec classes.

Table S5. Low-Shot Anomaly Detection Performance on MVTec [5], as measured on AUROC.

↓ Method \Shots →	1	2	5	10	16	20	50
Retained %	0.4	0.8	2.1	4.1	6.6	8.3	21
IMAGE-LEVEL AUROC							
SPADE	71.6 ± 0.7	73.4 ± 1.3	75.2 ± 1.5	77.5 ± 1.1	78.9 ± 0.9	79.6 ± 0.8	81.1 ± 0.4
PaDiM	76.1 ± 0.4	78.9 ± 0.6	81.0 ± 0.2	83.2 ± 0.7	85.5 ± 0.6	86.5 ± 0.3	90.1 ± 0.3
DifferNet	-	-	-	-	87.3	-	-
PatchCore-10	83.4 ± 0.6	86.4 ± 0.9	90.8 ± 0.8	93.6 ± 0.6	95.4 ± 0.7	95.8 ± 0.6	97.5 ± 0.3
PatchCore-25	84.1 ± 0.7	87.2 ± 1.0	91.0 ± 0.9	93.8 ± 0.5	95.5 ± 0.6	95.9 ± 0.6	97.7 ± 0.4
PIXEL-LEVEL AUROC							
SPADE	91.9 ± 0.3	93.1 ± 0.2	94.5 ± 0.1	95.4 ± 0.1	95.7 ± 0.2	95.7 ± 0.2	96.2 ± 0.0
PaDiM	88.2 ± 0.3	90.5 ± 0.2	92.5 ± 0.1	93.9 ± 0.1	94.8 ± 0.1	95.1 ± 0.1	96.3 ± 0.0
PatchCore-10	92.0 ± 0.2	93.1 ± 0.2	94.8 ± 0.1	96.2 ± 0.1	96.8 ± 0.3	96.9 ± 0.3	97.8 ± 0.0
PatchCore-25	92.4 ± 0.3	93.3 ± 0.2	94.8 ± 0.1	96.1 ± 0.1	96.8 ± 0.3	96.9 ± 0.3	97.7 ± 0.0
PRO METRIC							
SPADE	83.5 ± 0.4	85.8 ± 0.1	88.3 ± 0.2	89.6 ± 0.1	90.1 ± 0.2	90.1 ± 0.3	90.8 ± 0.1
PaDiM	72.4 ± 1.2	77.8 ± 0.7	82.7 ± 0.2	85.9 ± 0.2	87.5 ± 0.2	88.2 ± 0.2	90.4 ± 0.1
PatchCore-10	82.4 ± 0.3	85.1 ± 0.3	88.7 ± 0.2	90.9 ± 0.1	91.8 ± 0.2	92.0 ± 0.2	93.0 ± 0.1
PatchCore-25	83.7 ± 0.5	86.0 ± 0.3	88.8 ± 0.2	90.9 ± 0.1	91.7 ± 0.1	91.9 ± 0.2	92.8 ± 0.0

Table S6. Anomaly Detection Performance on MVTec [5], as measured on AUROC.

↓ Backbone	% of $\mathcal{M}$	Img. AUROC	Pw. AUROC	PRO
ResNet50 [23]	10	99.0	98.1	93.3
	1	98.7	97.8	93.3
WideResNet50 [57]	10	98.9	98.1	**93.5**
	1	99.0	98.0	93.1
ResNet101 [23]	10	98.6	97.9	92.5
	1	98.7	97.8	92.2
WideResNet101 [57]	10	99.1	**98.2**	93.4
	1	99.0	98.1	93.0
ResNeXt101 [55]	10	98.9	98.0	92.8
	1	98.7	97.8	92.6

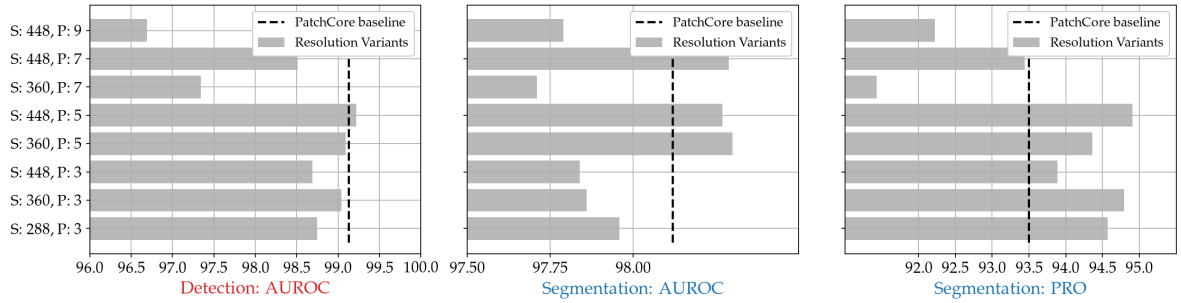

Figure S4. Influence of image size (S) and neighbourhood size (P) on *PatchCore* performance. The *PatchCore* baseline with default values is included for reference.

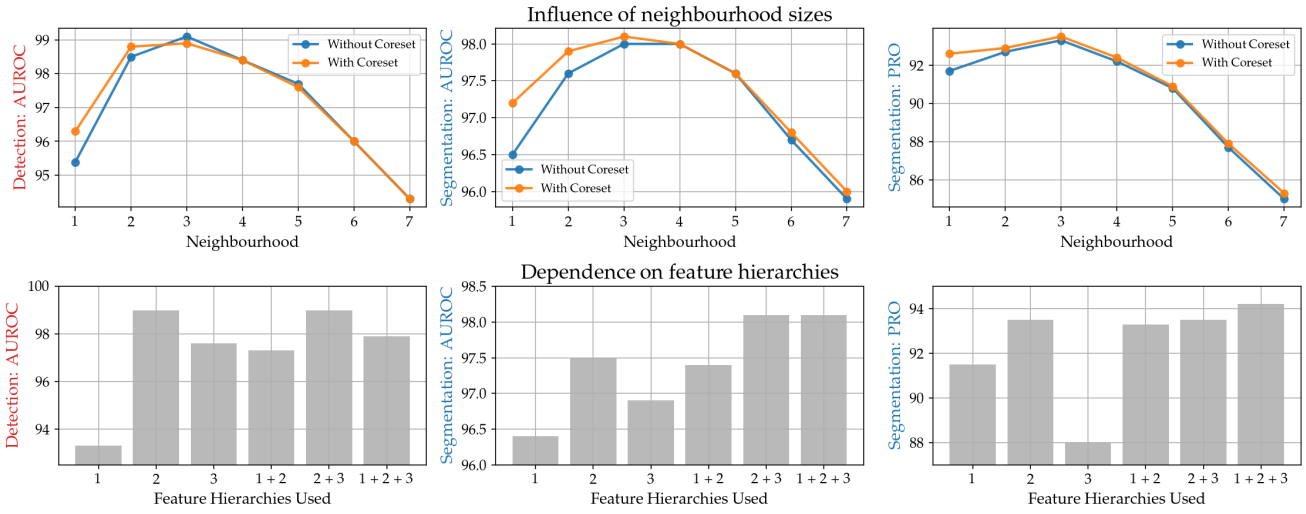

Figure S5. Influence of local awareness and network feature depths on anomaly detection performance.

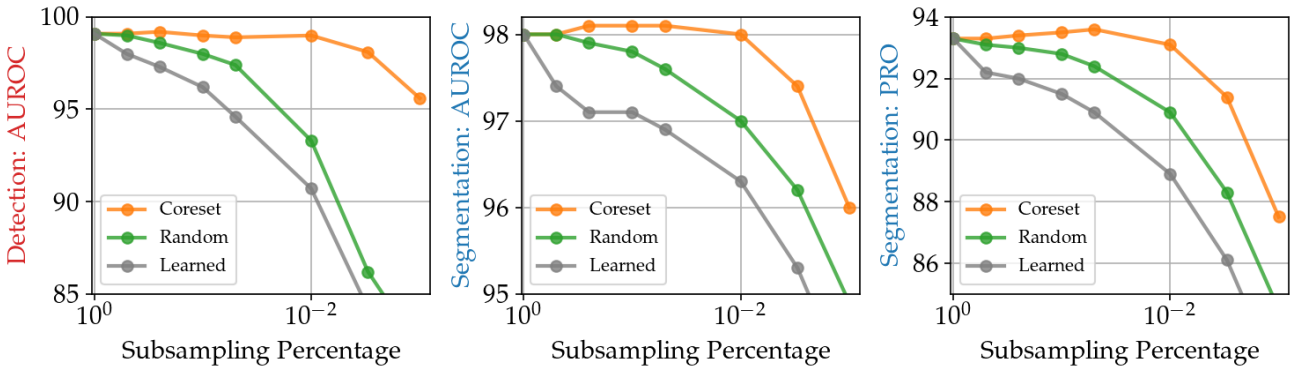

Figure S6. Performance retention for different subsamplers.