Test Time Training for Industrial Anomaly Segmentation

Alex Costanzino* Pierluigi Zama Ramirez*
Mirko Del Moro Agostino Aiezzo
Giuseppe Lisanti Samuele Salti Luigi Di Stefano
CVLAB, Department of Computer Science and Engineering (DISI) – University of Bologna, Italy

Abstract

Anomaly Detection and Segmentation (AD&S) is crucial for industrial quality control. While existing methods excel in generating anomaly scores for each pixel, practical applications require producing a binary segmentation to identify anomalies. Due to the absence of labeled anomalies in many real scenarios, standard practices binarize these maps based on some statistics derived from a validation set containing only nominal samples, resulting in poor segmentation performance. This paper addresses this problem by proposing a test time training strategy to improve the segmentation performance. Indeed, at test time, we can extract rich features directly from anomalous samples to train a classifier that can discriminate defects effectively. Our general approach can work downstream to any AD&S method that provides an anomaly score map as output, even in multimodal settings. We demonstrate the effectiveness of our approach over baselines through extensive experimentation and evaluation on MVTec AD and MVTec 3D-AD.

1. Introduction

Industrial Anomaly Detection (AD) aims to identify images that display samples with defects. A related commonly investigated task is Anomaly Segmentation (AS). Given an image of an anomalous sample, the goal is to identify the pixels corresponding to the defects. The latter is helpful for various practical applications, such as repairing only the damaged part of an object. Most existing methods [27] address both tasks together (AD&S), producing a map with anomaly scores for each image pixel as output (see Fig. 1, *Anomaly Score*). However, for practical applications, the anomaly scores must be binarized, *i.e.*, classifying each pixel as anomalous or nominal. This is supported by the ground truth of the AS task provided as a binary segmentation map (see Fig. 1, GT).

Collecting anomalous data, especially annotated ones, is expensive and time-consuming. Hence, the typical AD&S setup involves using only nominal samples during training. For this reason, as a standard practice [21], binarizing anomaly scores into anomaly maps concerns calculating a threshold by analyzing the score statistics on a validation set consisting solely of nominal samples. Subsequently, this threshold is applied during testing to binarize anomaly scores (see Fig. 1, column *THR*). However, there are no guarantees that this threshold is general enough to segment anomalies effectively at test time. Moreover, the threshold is often specific to each object class, thus needing frequent recalibration. Consequently, inaccurate anomaly segmentations may result even when the score map seems to highlight anomalies correctly. The availability of anoma-

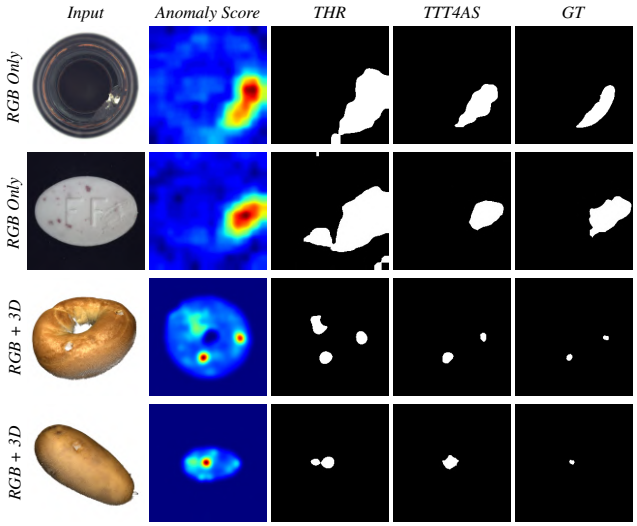

Figure 1. **Binary anomaly segmentation maps of anomalous samples.** Our approach, *TTT4AS*, enhances the quality of the binary anomaly segmentation masks. *TTT4AS* can be applied downstream to any anomaly detection and segmentation method that provides an anomaly score. The column THR represents the output of our baseline, a binarization obtained by computing a threshold based on the score statistics on a validation set, which contains only nominal samples.

*These authors contributed equally to this work.

lous samples would lead to better segmentation. Although this data might be obtainable at test time, it is not exploited by any current AD&S method.

The concept of utilizing test data to tailor the model specifically for a given scenario has gained traction in the literature in recent years. This approach is termed Test-Time Training (TTT) [42]. It is commonly employed to address domain-shift issues across various tasks, including classification [43] and semantic segmentation [22], seeking to adapt features learned from a deep model on a training set for a target test set. We are the first to explore TTT in the context of unsupervised AD&S. In this scenario, TTT holds the potential to be very effective as it allows us to utilize anomaly information that would not be available at training time. Furthermore, TTT for anomaly detection differs from TTT for classification and semantic segmentation tasks, making it an intriguing research problem. Specifically, as we can obtain highly informative features even for anomalous data by employing pre-trained and general-purpose feature extractors, such as DINO-v2 [32], TTT in this scenario concerns training a classifier for a new task rather than adapting a model on a new dataset. Indeed, we transition from a one-class classification problem (where only nominal samples are considered) to a binary classification problem (where both nominal and anomalous samples are exploited).

Our approach, named Test Time Training for Anomaly Segmentation (*TTT4AS*), operates downstream of any AD&S method that yields an anomaly score. It exploits the anomaly score to generate pseudo-annotations for selected nominal and anomalous pixels. Subsequently, we obtain a rich representation of anomalies and nominal data by extracting features for these pixel locations using general-purpose networks. We argue that anomaly features exhibit local similarity, enabling a classifier trained on even sparse points to effectively classify other features extracted from areas within the same test example. In practice, our method trains a simple ad-hoc SVM classifier for a test example. This classifier, leveraging information from anomalies, achieves superior segmentations than those obtained by binarizing the anomaly score maps solely by leveraging a threshold computed on nominal data only (see Fig. 1, column *TTT4AS* vs *THR*). Moreover, as our method involves training a simple SVM classifier, it limits the computational overhead during inference. Finally, our general approach can be applied downstream to any anomaly detection algorithm that provides an anomaly score, enhancing its segmentation quality. We apply it to RGB-based and multimodal (RGB + Point Cloud) methods to demonstrate its generality, yielding significant improvements on MVTec AD and MVTec 3D-AD datasets.

Briefly, our contributions are:

- For the first time, we investigate the TTT in $AD\&S$;
- We present a novel method, *TTT4AS*, for refining segmentation maps given an anomaly score generated by a generic AD&S algorithm and features extracted by a general-purpose pre-trained network;
- Our method enhances binary anomaly segmentation performance across various RGB or multimodal AD&S approaches. Moreover, our approach circumvents the need for selecting a specific threshold to binarize anomaly scores.

2. Related Works

Unsupervised AD&S. In recent years, many unsupervised AD&S approaches [27] have been proposed. A first kind of approaches detects anomalies by learning to reconstruct images containing nominal samples using various strategies such as auto-encoders [1, 18, 37, 55], inpainting methods [34], or diffusion models [48]. During testing, the trained model fails to reconstruct anomalous images, thus an anomaly score map can be generated by analyzing the differences between the input and reconstructed image on a per-pixel basis. A second type of approaches exploits features extracted by neural networks [8, 11, 15, 25, 28, 35, 36, 39, 41, 49–53, 56]. Several feature-based techniques follow the teacher-student paradigm [3, 5, 12, 40, 44] in which a teacher model extracts features from nominal samples and transfers its knowledge to a student model. Other approaches cast AD&S as a One Class Classification (OCC) problem, in which a classifier is trained to detect nominal samples using unsupervised techniques [35, 41, 51, 56] or by generating synthetic anomalies [25, 28, 50, 52]. Recently, the availability of powerful pre-trained feature extractors trained on large datasets (e.g., ImageNet [13], SA-1B [24]) in a supervised or self-supervised manner (e.g., [6, 16, 32]), has sparked interest resulting in new AD&S methods that utilize features obtained from these models. The underlying idea of these techniques, of which PatchCore [38] is one of the representatives, is to create a memory bank of features extracted from nominal samples during the training phase. During the inference phase, the features extracted from the test sample are compared with those stored in the memory bank to identify anomalies.

In recent years, the emergence of new multimodal benchmarks like MVTec 3D-AD [4] has led to the development of multimodal approaches that leverage both RGB images and 3D data. These methods aim to improve the reliability and efficiency of AD&S. Drawing inspiration from PatchCore [38], BTF [17] and M3DM [47] explores the application of memory banks for multimodal AD&S. The authors suggest incorporating 3D features, handcrafted [17] or learned by deep networks [33], alongside the 2D features obtained from a pre-trained network to enhance anomaly detection performance. CMM [10] achieves the highest per-

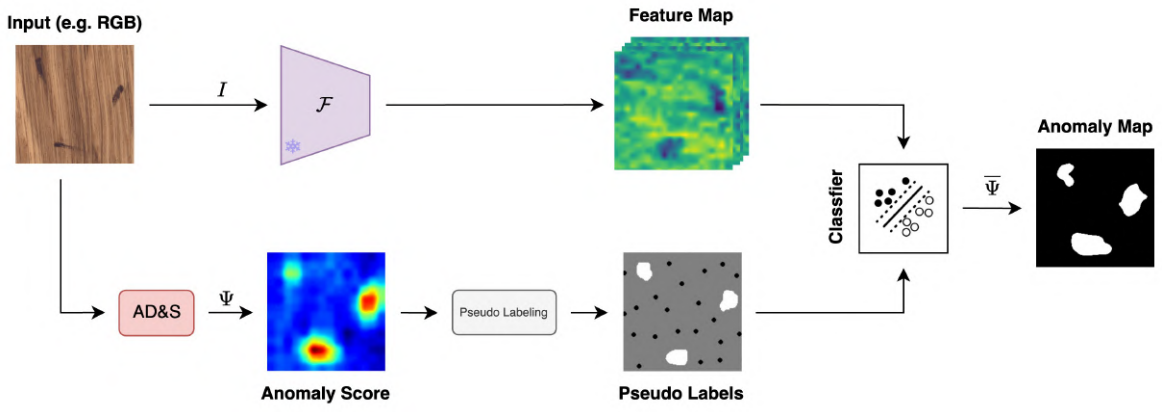

Figure 2. ***TTT4AS* Overview.** Given a single test input I such as an RGB image, a feature extractor $\mathcal{F}$ extracts a feature map, while an AD&S method predicts an anomaly score map Ψ. Then, exploiting Ψ, we create pseudo-labels for a sparse subset of points. Pseudo-labels and the corresponding features are employed as training data for an SVM Classifier. Finally, the trained SVM processes the dense feature map of the same test sample to predict a binary anomaly map $\overline{\Psi}$.

formance in this benchmark, proposing a novel cross-modal feature mapping paradigm.

Most of the aforementioned AD&S methods produce a pixel-wise anomaly score as output that can be employed to estimate a binary anomaly segmentation mask. We propose instead a general approach based on TTT that can be applied downstream to any AD&S technique improving its segmentation performance.

Test Time Training. Typically, once a model has been trained and deployed, it undergoes no further modifications. Nevertheless, following the pioneering work TTT [42], some methodologies have attempted to break this paradigm by leveraging unlabeled data available at test time to adapt models directly to the deployment scenario. Test-time training approaches have been applied in various computer vision tasks, including classification [29], object detection [23], or semantic segmentation [9], particularly to address the domain shift problem. These methods can be categorized based on their constraints on training and test data availability during training. Most approaches address the scenario where only test data is accessible for adaptation [14, 19, 26, 43], while others consider accessibility also of the training dataset [46]. Some techniques assume all test data is available for training the model [14, 19, 26, 29, 43], while others propose a real-time adaptation scenario where test data are obtained continuously one image at a time [9, 30, 31]. Few tackle the more challenging scenario of adapting to a single test image [20, 22]. Similarly to these latter methods, we train a new model on each test sample, yet we train only a new classifier rather than adapting an entire deep network. Moreover, our approach applies TTT in the context of AD&S for the first time.

3. Method

Problem Setup and Overview. Given an anomalous sample, the Anomaly Segmentation task aims to assign a binary label to each anomalous or nominal point.

Our approach, *TTT4AS*, is a downstream tool to any existing AD&S method that produces an anomaly score map to obtain better segmentation maps exploiting the anomaly information unavailable at training time. Given the anomaly score Ψ of a test sample, an input anomalous data I, and a frozen general-purpose feature extractor $\mathcal{F}$, our idea is to train, at test time, a specific classifier for each possible I. The classifier inputs are the feature extracted by $\mathcal{F}$ on some sparse points of I, while the supervision for these points is extracted from Ψ as *pseudo-labels*. Finally, the trained classifier is applied to the entire feature map, $\mathcal{F}(I)$, obtaining a dense anomaly binary map $\overline{\Psi}$. An overview of the entire framework is depicted in Fig. 2.

Anomaly Score. Our approach relies on a given AD&S method that produces an anomaly score Ψ as output. We are agnostic on the underlying mechanism, as evidenced by our experimental results in Sec. 4 where we employed diverse types of machinery and on different AD&S scenarios. Ψ is of dimension (H, W) and has values in an arbitrary range such as $[0, +\infty)$. We assume to find scores higher on anomalous points. These anomaly scores are typically converted using thresholds calculated on a validation set of nominal instances. In our case, however, they are utilized to obtain pseudo-labels for both anomalies and nominals, which are then used to train a classifier specific to each example.

Feature Extraction. Our objective is to train a classifier to segment anomalies effectively. An idea would be to utilize the anomaly score as input. However, this would not pro-

vide additional knowledge regarding anomalies (see Sec. 4 for more details). Information on these areas would be precious as they were unavailable during training. Our idea relies on the recent development of general-purpose feature extractors such as DINOv2 [32] and PointMAE [57] trained on large external datasets. These networks allow us to obtain rich features for both nominal and anomalous samples. These characteristics can be used as input to our classifier to boost segmentation performance. Thus, the next step in our pipeline consists of applying a general-purpose feature extractor $\mathcal{F}$ on the input I, obtaining a feature map with dimensions $H_f \times W_f \times D_f$. To obtain a feature associated with each location of the anomaly score map, we apply a bilinear upsample to obtain features with dimension $H \times W \times D_f$, namely $\overline{F}$. We highlight that we keep $\mathcal{F}$ frozen and never backpropagate gradient through it.

Pseudo-label Selection. Each sample must be associated with a label representing anomalous and nominal pixels to train the classifier. A naive approach would be to binarize the anomaly score with a threshold and then use it as supervision. However, this requires choosing such a threshold, which is not trivial due to the absence of anomalous samples during training, and it would also lead to many noisy labels. Ideally, to train a robust classifier on sparse supervision, we would like to obtain precise pseudo-labels. Yet, simultaneously, we would like to gather annotations for points associated with discriminative features for each anomaly in the object. In particular, we would like to retain as many as *hard* points as possible, e.g., the anomaly points with low anomaly scores. To achieve this objective, we first find all local maxima by comparing neighboring values (see Fig. 3 top right). Then, we keep only the peaks that exhibit an anomaly value higher than the i^{th} percentile of the anomaly score values (we set $i = 99$ in our experiments) and label these points as anomalous (see Fig. 3) bottom left). We call these points as *easy* pseudo-labels, since they are already associated with high anomaly score values. However, to obtain a good classifier, we need to include also the *hard* samples mentioned above. Hence, we add the spatial neighboring points of the non-suppressed peaks (see Fig. 3 bottom right). These points might have lower anomaly scores yet are likely to belong to an anomaly as they are spatially close to the peak. Regarding nominal pseudo labels, we spatially uniformly sample the remaining points. In this way, we collect both *easy* and *hard* nominal points. Moreover, as they often occupy most of the image, we will only have a limited amount of noisy labels.

Test Time Training. At this point, we hold a reasonable set of sparse pseudo labels, $\overline{\psi}$, corresponding to both *easy* and *hard* samples for both classes and rich features for these points, f. Thus, we can train the classifier on a single anomalous test sample exploiting this data (see Fig. 4). Therefore, we optimize a soft-margin SVM classifier [45]

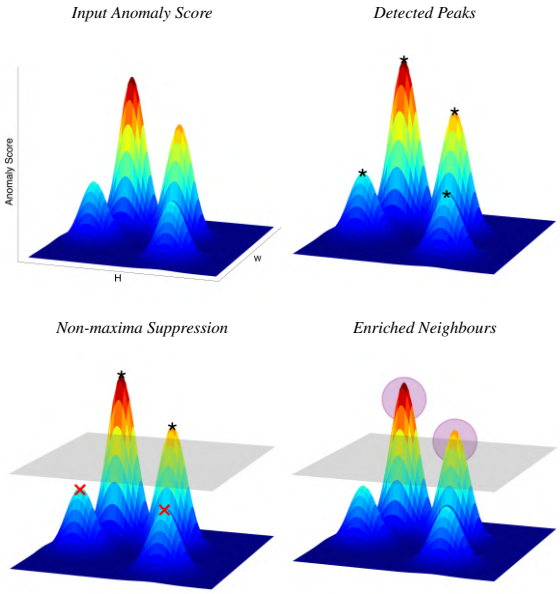

Figure 3. **Pseudo-labels Selection.** Starting from an anomaly score map (top left) all local maxima are computed by neighbouring values comparison (top right). Then, the peaks above a certain percentile (gray plane, bottom left) are kept while the others are suppressed. Finally, the non-suppressed maxima are enriched with their spatial neighbouring points (purple spheres, bottom right) and labeled as anomalous.

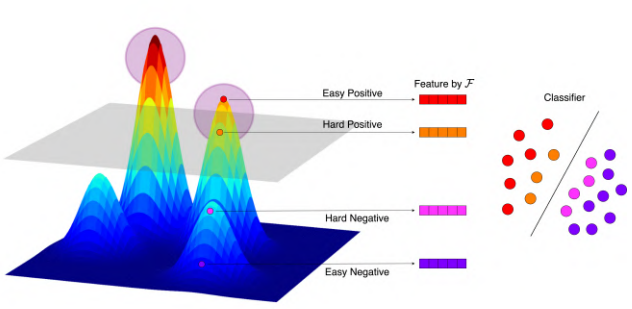

Figure 4. **Test Time Training.** The binary classifier is trained on both *easy* and *hard* samples for both classes, retrieved thanks to the aforementioned pseudo-labeling procedure.

by minimizing the Hinge Loss between f and $\overline{\psi}$:

$$\mathcal{L} = \|w\|^2 + C\left[\frac{1}{n}\sum_{i=1}^{n}\max\left(0, 1 - \overline{\psi}_i(w^\top f_i - b)\right)\right] \quad (1)$$

in which the parameters w, b represent respectively the weights and biases of the SVM classifier, n represents the number of training data, while C represents a regularization factor. After the optimization, we can predict a dense binary

Metric	*Bottle*	*Cable*	*Capsule*	*Carpet*	*Grid*	*Hazelnut*	*Leather*	*Metal Nut*	*Pill*	*Screw*	*Tile*	*Toothbrush*	*Transistor*	*Wood*	*Zipper*	**Mean**
	(a) PatchCore [38] with WideResnet-50 [54] - Anomaly Score															
I-AUROC	1.000	0.956	0.951	0.983	0.929	1.000	1.000	0.983	0.920	0.958	0.988	0.967	0.998	0.987	0.987	0.974
P-AUROC	0.978	0.974	0.983	0.983	0.964	0.981	0.984	0.962	0.987	0.984	0.940	0.980	0.973	0.920	0.976	0.971
	(b) PatchCore [38] with WideResnet-50 [54] - Binary Map - *THR* with $\mu + 2\sigma$															
Precision	0.397	0.344	0.278	0.362	0.432	0.405	0.297	0.435	0.347	0.298	0.403	0.286	0.334	0.384	0.268	0.351
Recall	0.510	0.465	0.626	0.522	0.428	0.380	0.542	0.566	0.618	0.522	0.517	0.542	0.287	0.469	0.605	0.507
F1 Score	0.175	0.194	0.085	0.092	0.078	0.120	0.045	0.311	0.188	0.066	0.209	0.123	0.114	0.121	0.119	0.136
	(c) PatchCore [38] with WideResnet-50 [54] - Binary Map - *THR* with $\mu + 3\sigma$															
Precision	0.285	0.495	0.29	0.334	0.464	0.218	0.31	0.393	0.338	0.3	0.468	0.216	0.389	0.304	0.267	0.338
Recall	0.601	0.603	0.693	0.603	0.459	0.457	0.655	0.585	0.614	0.59	0.634	0.557	0.471	0.519	0.702	0.583
F1 Score	0.236	0.326	0.117	0.118	0.109	0.176	0.053	0.401	0.279	0.115	0.286	0.191	0.21	0.15	0.173	0.196
	(d) PatchCore [38] with WideResnet-50 [54] - Binary Map - *THR* with $\mu + 4\sigma$															
Precision	0.173	0.280	0.194	0.157	0.278	0.133	0.127	0.344	0.261	0.169	0.307	0.138	0.149	0.173	0.186	0.205
Recall	0.504	0.365	0.566	0.501	0.314	0.330	0.532	0.427	0.419	0.409	0.496	0.343	0.255	0.415	0.579	0.430
F1 Score	0.231	0.244	0.128	0.118	0.110	0.170	0.053	0.35	0 0.268	0.124	0.253	0.163	0.139	0.142	0.185	0.179
	(e) PatchCore [38] with WideResnet-50 [54] - Binary Map - *TTT4AS*															
Precision	0.693	0.506	0.192	0.327	0.132	0.308	0.201	0.561	0.281	0.086	0.582	0.205	0.448	0.364	0.498	**0.359**
Recall	0.612	0.549	0.804	0.664	0.626	0.760	0.795	0.463	0.700	0.771	0.408	0.443	0.541	0.43	0.589	**0.610**
F1 Score	0.564	0.446	0.245	0.333	0.196	0.349	0.260	0.388	0.296	0.149	0.386	0.211	0.345	0.31	0.453	**0.329**

Table 1. **Performance on MVTec AD dataset [2] with PatchCore [38] trained on Wide ResNet-50 features [54].** Best results in **bold**.

Metric	*Bottle*	*Cable*	*Capsule*	*Carpet*	*Grid*	*Hazelnut*	*Leather*	*Metal Nut*	*Pill*	*Screw*	*Tile*	*Toothbrush*	*Transistor*	*Wood*	*Zipper*	**Mean**
	(a) PatchCore [38] with DINO-v2 [32] - Anomaly Score															
I-AUROC	1.000	0.949	0.895	0.997	1.000	1.000	1.000	0.993	0.949	0.857	0.994	1.000	0.965	0.975	0.998	0.971
P-AUROC	0.978	0.914	0.962	0.986	0.971	0.989	0.971	0.973	0.979	0.804	0.941	0.982	0.967	0.890	0.938	0.950
	(b) PatchCore [38] with DINO-v2 [32] - Binary Map - *THR* with $\mu + 3\sigma$															
Precision	0.268	0.297	0.231	0.143	0.068	0.206	0.115	0.35	0.289	0.143	0.253	0.178	0.31	0.161	0.23	0.216
Recall	0.603	0.54	0.633	0.619	0.513	0.464	0.657	0.629	0.681	0.118	0.643	0.712	0.436	0.52	0.671	0.563
F1 Score	0.139	0.236	0.093	0.072	0.026	0.162	0.026	0.353	0.182	0.054	0.161	0.108	0.219	0.16	0.167	0.144
	(c) PatchCore [38] with DINO-v2 [32] - Binary Map - *TTT4AS*															
Precision	0.662	0.502	0.163	0.413	0.185	0.425	0.212	0.644	0.337	0.046	0.644	0.272	0.391	0.47	0.449	**0.388**
Recall	0.664	0.565	0.632	0.824	0.787	0.861	0.893	0.528	0.74	0.361	0.495	0.594	0.462	0.664	0.644	**0.648**
F1 Score	0.593	0.48	0.197	0.457	0.272	0.499	0.286	0.482	0.358	0.078	0.474	0.301	0.318	0.464	0.469	**0.382**

Table 2. **Performance on MVTec AD dataset [2] with PatchCore [38] trained on DINO-v2 features [32].** Best results in **bold**.

anomaly map $\overline{\Psi}$ given the dense feature map $\overline{F}$. We opted for a simple SVM classifier due to its fast training property. We believe that this procedure is effective because the selected sparse points with pseudo-labels adequately represent the anomalous and nominal parts. Consequently, when applied to the dense input, the classifiers can cluster the features into the two classes.

4. Experimental Settings

Implementation Details. We employ both Convolutional and Transformers-based backbones to realize the feature extractor, $\mathcal{F}$, *i.e*. WideResNet-50 [54] trained on ImageNet [13], DINO [6] trained on ImageNet [13], DINO-v2 [32] trained on a large and diverse dataset of 142 million images, and Point-MAE [33, 57] trained on ShapeNet [7]. We implement the SVM classifier with LinearSVC from *scikit-learn* with a margin regularization factor C of 0.001. To evaluate the generality of our framework, we apply it downstream to both RGB-only [38] and multi-modal (RGB+3D) AD&S methods [10, 47]. Moreover, we consider approaches that estimate the anomaly scores with different strategies, *i.e*. PatchCore [38] and M3DM [47] which are based on memory-banks, while CFM [10] is a feature-reconstruction method. We conducted experiments using the original code from the authors of the AD&S methods when available or re-implemented otherwise. We implement the PatchCore and M3DM memory banks with a coreset subsampling ratio of 10% and a random projection radius of 0.9. All the experiments run on a single NVIDIA GeForce RTX 4090.

Benchmarks. We evaluate our framework on two AD&S benchmarks. MVTec AD [2] is an RGB-only dataset that contains 15 categories with 5354 images, 1725 of which are in the test set. Each class is divided into nominal-only training data and test sets containing nominal and anomalous samples for a specific product with various defect types and anomaly ground truth masks. Since no validation set is provided, we reserve for each class a 20% split from the training data to be used as validation data. Following the common practice [38], images are resized and center cropped to 256×256 and 224×224, respectively. MVTec 3D-AD [4] is a multi-modal dataset comprising 10 categories of industrial objects, totaling 2656 train samples, 294 validation samples, and 1197 test samples. This dataset provides RGB images alongside pixel-registered 3D information for each sample. Thus, we have RGB information at each pixel location paired with (x, y, z) coordinates. Following the common practice, images and 3D data are resized to 224×224. Moreover, as done in [10, 17, 47],

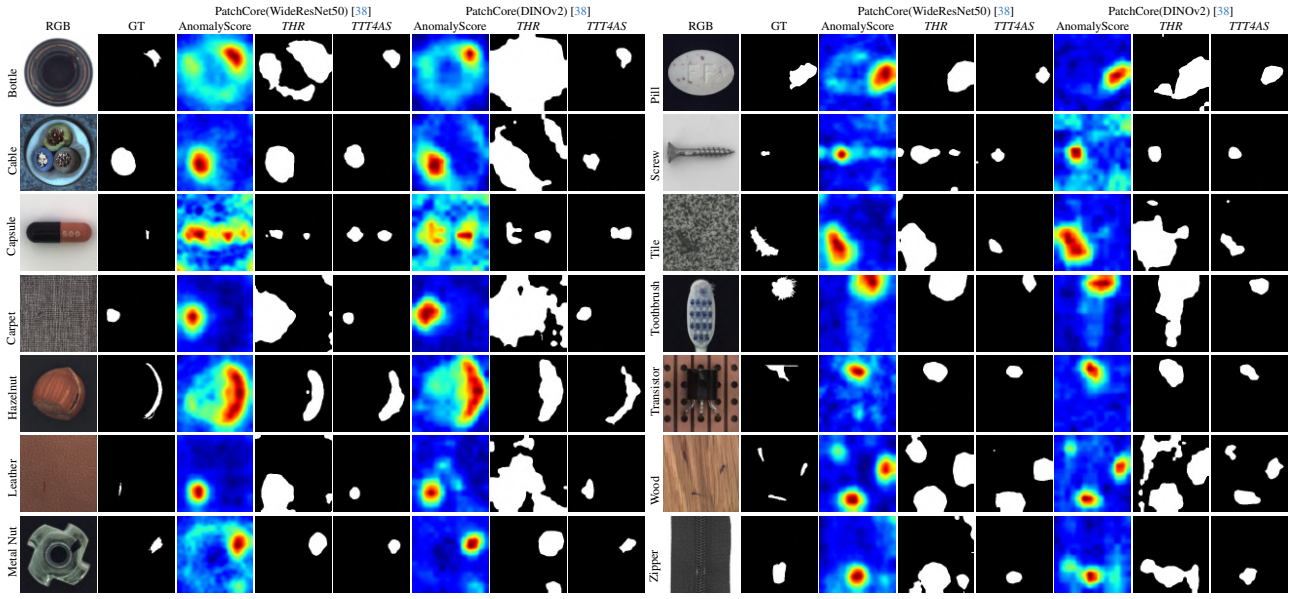

Figure 5. **MVTec AD Qualitative Results.** We show for each class: RGB, ground truth followed by anomaly score, binary segmentation maps with thresholding, binary segmentation maps with *TTT4AS* for PatchCore [38] with backbone WideResNet50 [54] and DINO-v2 [32]

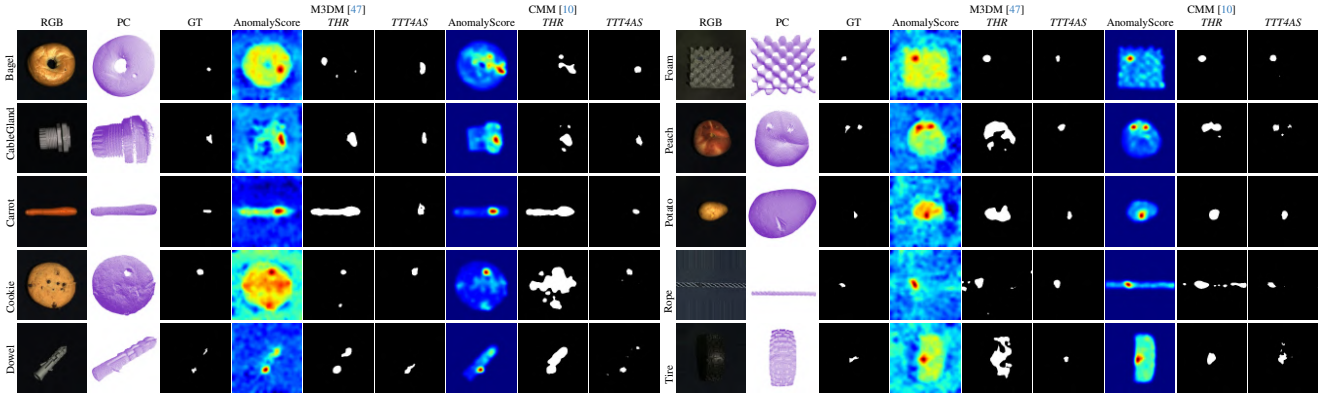

Figure 6. **MVTec-3D AD Qualitative Results.** we show for each class: RGB, point cloud, ground truth followed by anomaly score, binary segmentation maps with thresholding, binary segmentation maps with *TTT4AS* for both M3DM [47] and CMM [10].

we fit a plane with RANSAC on the 3D data and consider a point as background if the distance to the plane is less than 0.005. In both cases, no data augmentation is applied, as this would require prior knowledge about class-retaining augmentations [38].

Metrics. We assess the performance for the binary anomaly segmentation task by computing the pixel-level Precision, Recall, and F1 Score only on samples containing anomalous areas. However, since most of these methods tend to produce a loose binary anomaly map that saturates the Recall, we use the F1 Score as the primary metric to rank our experiments. Moreover, as a reference, we also report the MVTec benchmark [2, 4] metrics for Anomaly Detection, i.e., the Area Under the Receiver Operator Curve (I-AUROC) computed on the global anomaly score, and the pixel-level Area Under the Receiver Operator Curve (P-AUROC). As this latter metric evaluates scores rather than a discrete segmentation map, we do not use it to evaluate the effectiveness of our methodology.

5. Experimental Results

Baseline. We employ as baselines the binary anomaly maps obtained by thresholding anomaly scores produced by AD&S methods based on statistics computed on the validation set, which contains only nominal samples. In particular, we calculate the mean μ and the standard deviation σ of each pixel of anomaly scores obtained for each sample in the validation set. Then, at test time, following the stan-

Method	*Bagel*	*Cable Gland*	*Carrot*	*Cookie*	*Dowel*	*Foam*	*Peach*	*Potato*	*Rope*	*Tire*	**Mean**
	(a) M3DM [47] - Anomaly Score										
I-AUROC	0.994	0.909	0.972	0.976	0.960	0.942	0.973	0.899	0.972	0.850	0.945
P-AUROC	0.995	0.993	0.997	0.985	0.985	0.984	0.996	0.994	0.997	0.996	0.992
	(b) M3DM [47] - Binary Map - *THR* with $\mu + 3\sigma$										
Precision	0.174	0.105	0.045	0.493	0.221	0.254	0.067	0.050	0.194	0.127	0.173
Recall	0.949	0.980	0.997	0.712	0.909	0.536	1.000	0.999	0.917	0.894	**0.889**
F1 Score	0.270	0.174	0.085	0.547	0.328	0.318	0.121	0.094	0.308	0.204	0.245
	(c) M3DM [47] - Binary Map - *TTT4AS*										
Precision	0.498	0.486	0.337	0.752	0.464	0.386	0.536	0.347	0.561	0.302	**0.467**
Recall	0.607	0.706	0.750	0.351	0.691	0.624	0.779	0.684	0.543	0.669	0.640
F1 Score	0.478	0.525	0.422	0.443	0.514	0.44	0.585	0.419	0.468	0.383	**0.468**

Table 3. **Performance on MVTec 3D-AD dataset [4] with M3DM [47].** Best results in **bold**.

Method	*Bagel*	*Cable Gland*	*Carrot*	*Cookie*	*Dowel*	*Foam*	*Peach*	*Potato*	*Rope*	*Tire*	**Mean**
	(a) CMM [10] - Anomaly Score										
I-AUROC	0.994	0.888	0.984	0.993	0.980	0.888	0.941	0.943	0.980	0.953	0.954
P-AUROC	0.997	0.992	0.999	0.972	0.987	0.993	0.998	0.999	0.998	0.998	0.993
	(b) CMM [10] - Binary Map - *THR* with $\mu + 3\sigma$										
Precision	0.301	0.188	0.049	0.518	0.072	0.275	0.262	0.092	0.049	0.182	0.198
Recall	0.949	0.842	0.998	0.901	0.896	0.597	0.957	0.998	0.989	0.896	**0.902**
F1 Score	0.425	0.265	0.092	0.619	0.129	0.327	0.375	0.160	0.091	0.267	0.275
	(c) CMM [10] - Binary Map - *TTT4AS*										
Precision	0.432	0.258	0.242	0.713	0.195	0.214	0.353	0.252	0.264	0.111	**0.303**
Recall.	0.745	0.766	0.889	0.603	0.739	0.732	0.872	0.888	0.865	0.904	0.800
F1 Score	0.495	0.362	0.351	0.606	0.289	0.311	0.470	0.363	0.360	0.189	**0.380**

Table 4. **Performance on MVTec 3D-AD dataset [4] with CMMs [10].** Best results in **bold**.

dard practices [21], we consider anomalous all the points above a threshold equal to $\mu + c \cdot \sigma$ and nominal all the rest. As shown in Tab. 1, we evaluate this approach with several c on PatchCore [38] trained with WideResnet-50 [54] features (experiments (a), (b) and (c)), finding $\mu + 3\sigma$ as the optimal threshold, which is employed to create the baseline for all the experiments.

2D Anomaly Segmentation. We evaluate our proposal on MVTec AD dataset, reporting results in Tab. 1, Tab. 2. We trained PatchCore [38] with WideResnet-50 [54] features, as depicted in Tab. 1. Our method outperforms the baselines in all mean Precision, Recall, and F1 score metrics. In particular, we note an improvement of 2.1% in Precision, 2.7% in Recall, and a remarkable 15% in F1 Score ((e) - *TTT4AS* vs (c) - *THR* with $\mu + 3\sigma$). We then repeat the experiments on MVTec AD by training PatchCore with DINO-v2 features, as shown in Tab. 2. Once again, the method outperforms the baseline in all three mean metrics, in particular with an increase of 17.2% in Precision, 8.5% in Recall, and a 23.8% in F1 Score ((c) - *TTT4AS* vs (b) - *THR* with $\mu + 3\sigma$). It is worth highlighting that even though the baseline with PatchCore with WideResnet-50 features performs better than the baseline of PatchCore with DINO-v2 features ((c) in Tab. 1 vs (b) in Tab. 2), with our approach, we obtain better performance by employing DINO-v2 features ((e) in Tab. 1 vs (c) in Tab. 2). We argue that the higher expressiveness of DINO-v2 features allows the SVM classifier to distinguish anomalous and nominal pixels better. In Fig. 5, we show qualitative results on most classes of the MVTec dataset. Our method provides remarkably sharper and more accurate anomaly segmentations than the baseline. We also highlight that PatchCore creates the memory banks on the same features used by *TTT4AS* to train the SVM classifier. Nevertheless, we can still improve the performance remarkably.

Multi-Modal Anomaly Segmentation. To assess the generality of our approach, we apply *TTT4AS* also to multimodal methods. In particular, we apply *TTT4AS* to M3DM (Tab. 3) and CMM (Tab. 4) on the MVTec 3D-AD dataset. Our approach outperforms the baseline in terms of mean Precision and mean F1 Score with both methods. We note that though *TTT4AS* on top of M3DM decreases the Recall of 24.9%, it improves the Precision of 29.4% and the F1 Score of 22.3%. Also, with *TTT4AS* on top of CMM, we have an increase of 10.5% in Precision, a decrease of 10.2% in Recall, but an overall increase of 10.5% in F1 Score. The higher F1 Score indicates an overall improvement in the binary segmentation maps. In Fig. 6, we show qualitative results for all the classes of the MVTec 3D-AD dataset. Compared to the baseline, our method provides more accurate

Metric	*Bottle*	*Cable*	*Capsule*	*Carpet*	*Grid*	*Hazelnut*	*Leather*	*Metal Nut*	*Pill*	*Screw*	*Tile*	*Toothbrush*	*Transistor*	*Wood*	*Zipper*	**Mean**
	Percentile @ 99.5															
Precision	0.683	0.496	0.189	0.334	0.131	0.319	0.196	0.555	0.294	0.093	0.569	0.197	0.432	0.382	0.505	0.358
Recall	0.592	0.513	0.774	0.684	0.593	0.743	0.788	0.447	0.675	0.733	0.395	0.421	0.497	0.422	0.584	0.591
F1 Score	0.538	0.432	0.242	0.344	0.190	0.344	0.253	0.376	0.306	0.158	0.368	0.196	0.328	0.312	0.453	0.323
	Percentile @ 99.0															
Precision	0.693	0.506	0.192	0.327	0.132	0.308	0.201	0.561	0.281	0.086	0.582	0.205	0.448	0.364	0.498	0.359
Recall	0.612	0.549	0.804	0.664	0.626	0.760	0.795	0.463	0.700	0.771	0.408	0.443	0.541	0.43	0.589	0.610
F1 Score	0.564	0.446	0.245	0.333	0.196	0.349	0.260	0.388	0.296	0.149	0.386	0.211	0.345	0.31	0.453	0.329
	Percentile @ 98.0															
Precision	0.683	0.501	0.177	0.322	0.124	0.302	0.189	0.546	0.276	0.069	0.592	0.216	0.448	0.359	0.473	0.352
Recall	0.654	0.597	0.831	0.690	0.640	0.779	0.847	0.510	0.768	0.812	0.446	0.551	0.584	0.469	0.646	0.655
F1 Score	0.583	0.466	0.224	0.338	0.190	0.349	0.252	0.402	0.299	0.124	0.423	0.239	0.355	0.327	0.470	0.336
	Percentile @ 95.0															
Precision	0.622	0.452	0.137	0.259	0.103	0.270	0.132	0.485	0.239	0.044	0.585	0.197	0.396	0.346	0.368	0.309
Recall	0.746	0.743	0.895	0.833	0.746	0.829	0.916	0.625	0.843	0.921	0.542	0.740	0.666	0.573	0.811	0.762
F1 Score	0.603	0.500	0.185	0.330	0.168	0.340	0.202	0.401	0.268	0.083	0.479	0.259	0.338	0.361	0.460	0.332
	Percentile @ 90.0															
Precision	0.544	0.369	0.096	0.193	0.076	0.222	0.093	0.405	0.189	0.029	0.546	0.174	0.331	0.302	0.244	0.254
Recall	0.823	0.857	0.933	0.882	0.850	0.872	0.954	0.703	0.886	0.944	0.679	0.818	0.765	0.661	0.904	0.835
F1 Score	0.571	0.462	0.143	0.278	0.133	0.297	0.150	0.360	0.218	0.056	0.52	0.245	0.300	0.356	0.357	0.296

Table 5. **Performance on MVTec AD dataset with PatchCore trained on Wide ResNet-50 features with Pseudo-label Selection performed with different percentiles.**

Metric	*Bottle*	*Cable*	*Capsule*	*Carpet*	*Grid*	*Hazelnut*	*Leather*	*Metal Nut*	*Pill*	*Screw*	*Tile*	*Toothbrush*	*Transistor*	*Wood*	*Zipper*	**Mean**
	TTT4AS - WideResNet50 [54] features as input to classifier															
Precision	0.693	0.506	0.192	0.327	0.132	0.308	0.201	0.561	0.281	0.086	0.582	0.205	0.448	0.364	0.498	**0.359**
Recall	0.612	0.549	0.804	0.664	0.626	0.760	0.795	0.463	0.700	0.771	0.408	0.443	0.541	0.43	0.589	**0.610**
F1 Score	0.564	0.446	0.245	0.333	0.196	0.349	0.260	0.388	0.296	0.149	0.386	0.211	0.345	0.31	0.453	**0.329**
	TTT4AS - anomaly score as input to classifier															
Precision	0.594	0.477	0.205	0.293	0.185	0.272	0.189	0.388	0.251	0.134	0.440	0.155	0.414	0.324	0.545	0.324
Recall	0.499	0.404	0.678	0.637	0.602	0.488	0.781	0.330	0.501	0.598	0.287	0.266	0.467	0.268	0.667	0.498
F1 Score	0.456	0.387	0.263	0.336	0.238	0.314	0.255	0.312	0.276	0.187	0.285	0.172	0.356	0.240	0.547	0.308

Table 6. **Performance on MVTec AD dataset.** *TTT4AS* with WideResNet50 [54] features or anomaly score as input to the classifier. We employ PatchCore [38] trained on WideResNet-50 features [54] as AD&S method. Best results in **bold**.

anomaly segmentations.

Ablation Study We conducted an ablation study on the choice of the percentile employed to detect peaks during pseudo-labels generation (see Sec. 3). We argue that the choice of this threshold is not critical, since, as shown in Tab. 5, the performance is stable across a wide range of different percentiles, *i.e*. 99.5th, 99th, 98th and 95th, with a start in decreasing around the 90th percentile.

We also experimented with the importance of extracting features from the pre-trained backbones as input features for the SVM classifier. One could argue that, given the same feature selection algorithm for the binary pseudo-labels, using the values from the anomaly score as input features for the SVM could be sufficient. However, as shown in Tab. 6, this approach yields worse results than the case in which the input features provided to the SVM are extracted from pre-trained backbones.

6. Limitations & Conclusion

We have developed an effective method to segment anomalies given an anomaly score generated by an AD&S algorithm. We proved that this simple approach outperforms simple baselines through extensive experimentation and evaluation of benchmark datasets. Moreover, we have shown that this approach is general and can be applied downstream to many AD&S methods.

However, we are aware that our method presents several limitations. Firstly, the non-maxima suppression criterion is based on the percentile of the anomaly score. Therefore, sometimes, some good maxima might also be suppressed, leading to a mislabeling that is hardly recoverable even with the soft-margin SVM classifier properties. Nevertheless, Sec. 4 show that performance is stable across various different percentiles for the experimented datasets. Secondly, our approach assumes that we can collect several *hard* samples that serve as support vectors for the classifier from the spatial neighborhood of score peaks. Even though our heuristic is effective, other heuristics able to recover more and better samples may be developed. Thirdly, our approach requires additional computational overhead at test time. Even though we selected linear SVM as our classifier, which is relatively fast, our procedure still increased the inference time. We aim to address these limitations in future work and hope that the new idea of employing test time training for unsupervised industrial anomaly detection may inspire other researchers and foster further advances in the field.

References

[1] 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. 2

[2] Paul Bergmann, Michael Fauser, David Sattlegger, and Carsten Steger. Mvtec ad – a comprehensive real-world dataset for unsupervised anomaly detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 5, 6

[3] Paul Bergmann, Michael Fauser, David Sattlegger, and Carsten Steger. Uninformed students: Student-teacher anomaly detection with discriminative latent embeddings. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4183–4192, 2020. 2

[4] Paul Bergmann, Jin Xin, David Sattlegger, and Carsten Steger. The mvtec 3d-ad dataset for unsupervised 3d anomaly detection and localization. In *Proceedings of the 17th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*, pages 202–213, 2022. 2, 5, 6, 7

[5] Yunkang Cao, Qian Wan, Weiming Shen, and Liang Gao. Informative knowledge distillation for image anomaly segmentation. *Knowledge-Based Systems*, 248:108846, 2022. 2

[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 International Conference on Computer Vision (ICCV)*, 2021. 2, 5

[7] Angel X. Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and Fisher Yu. ShapeNet: An Information-Rich 3D Model Repository. Technical Report arXiv:1512.03012 [cs.GR], Stanford University — Princeton University — Toyota Technological Institute at Chicago, 2015. 5

[8] Li-Ling Chiu and Shang-Hong Lai. Self-supervised normalizing flows for image anomaly detection and localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2926–2935, 2023. 2

[9] Marc Botet Colomer, Pier Luigi Dovesi, Theodoros Panagiotakopoulos, Joao Frederico Carvalho, Linus Härenstam-Nielsen, Hossein Azizpour, Hedvig Kjellström, Daniel Cremers, and Matteo Poggi. To adapt or not to adapt? real-time adaptation for semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16548–16559, 2023. 3

[10] Alex Costanzino, Pierluigi Zama Ramirez, Giuseppe Lisanti, and Luigi Di Stefano. Multimodal industrial anomaly detection by crossmodal feature mapping. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024. 2, 5, 6, 7

[11] 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

[12] 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

[13] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 2, 5

[14] Sachin Goyal, Mingjie Sun, Aditi Raghunathan, and J Zico Kolter. Test time adaptation via conjugate pseudo-labels. *Advances in Neural Information Processing Systems*, 35:6204–6218, 2022. 3

[15] Denis Gudovskiy, Shun Ishizaka, and Kazuki Kozuka. Cflow-ad: Real-time unsupervised anomaly detection with localization via conditional normalizing flows. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 98–107, 2022. 2

[16] 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

[17] Eliahu Horwitz and Yedid Hoshen. Back to the feature: classical 3d features are (almost) all you need for 3d anomaly detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2967–2976, 2023. 2, 5

[18] Jinlei Hou, Yingying Zhang, Qiaoyong Zhong, Di Xie, Shiliang Pu, and Hong Zhou. Divide-and-assemble: Learning block-wise memory for unsupervised anomaly detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8791–8800, 2021. 2

[19] Yusuke Iwasawa and Yutaka Matsuo. Test-time classifier adjustment module for model-agnostic domain generalization. *Advances in Neural Information Processing Systems*, 34:2427–2440, 2021. 3

[20] Klara Janouskova, Tamir Shor, Chaim Baskin, and Jiri Matas. Single image test-time adaptation for segmentation. *arXiv preprint arXiv:2309.14052*, 2023. 3

[21] Zhiqian Jiang, Yu Zhang, Yong Wang, Jinlong Li, and Xiaorong Gao. Fr-patchcore: An industrial anomaly detection method for improving generalization. *Sensors*, 24(5), 2024. 1, 7

[22] Ansh Khurana, Sujoy Paul, Piyush Rai, Soma Biswas, and Gaurav Aggarwal. Sita: Single image test-time adaptation. *arXiv preprint arXiv:2112.02355*, 2021. 2, 3

[23] Junho Kim, Inwoo Hwang, and Young Min Kim. Ev-tta: Test-time adaptation for event-based object recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17745–17754, 2022. 3

[24] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollar, and Ross Girshick. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4015–4026, 2023. 2

[25] Chun-Liang Li, Kihyuk Sohn, Jinsung Yoon, and Tomas Pfister. Cutpaste: Self-supervised learning for anomaly detection and localization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9664–9674, 2021. 2

[26] Jian Liang, Dapeng Hu, and Jiashi Feng. Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation. In *International conference on machine learning*, pages 6028–6039. PMLR, 2020. 3

[27] Jiaqi Liu, Guoyang Xie, Jingbao Wang, Shangnian Li, Chengjie Wang, Feng Zheng, and Yaochu Jin. Deep industrial image anomaly detection: A survey. *arXiv preprint arXiv:2301.11514*, 2, 2023. 1, 2

[28] Fabio Valerio Massoli, Fabrizio Falchi, Alperen Kantarci, Şeymanur Akti, Hazim Kemal Ekenel, and Giuseppe Amato. Mocca: Multilayer one-class classification for anomaly detection. *IEEE Transactions on Neural Networks and Learning Systems*, 33(6):2313–2323, 2021. 2

[29] Zachary Nado, Shreyas Padhy, D Sculley, Alexander D'Amour, Balaji Lakshminarayanan, and Jasper Snoek. Evaluating prediction-time batch normalization for robustness under covariate shift. *arXiv preprint arXiv:2006.10963*, 2020. 3

[30] A Tuan Nguyen, Thanh Nguyen-Tang, Ser-Nam Lim, and Philip HS Torr. Tipi: Test time adaptation with transformation invariance. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24162–24171, 2023. 3

[31] Shuaicheng Niu, Jiaxiang Wu, Yifan Zhang, Yaofo Chen, Shijian Zheng, Peilin Zhao, and Mingkui Tan. Efficient test-time model adaptation without forgetting. In *International conference on machine learning*, pages 16888–16905. PMLR, 2022. 3

[32] Maxime Oquab, Timothée Darcet, Theo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Russell Howes, Po-Yao Huang, Hu Xu, Vasu Sharma, Shang-Wen Li, Wojciech Galuba, Mike Rabbat, Mido Assran, Nicolas Ballas, Gabriel Synnaeve, Ishan Misra, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision, 2023. 2, 4, 5, 6

[33] Yatian Pang, Wenxiao Wang, Francis EH Tay, Wei Liu, Yonghong Tian, and Li Yuan. Masked autoencoders for point cloud self-supervised learning. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part II*, pages 604–621. Springer, 2022. 2, 5

[34] Jonathan Pirnay and Keng Chai. Inpainting transformer for anomaly detection. In *International Conference on Image Analysis and Processing*, pages 394–406. Springer, 2022. 2

[35] Tal Reiss, Niv Cohen, Liron Bergman, and Yedid Hoshen. Panda: Adapting pretrained features for anomaly detection and segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2806–2814, 2021. 2

[36] Oliver Rippel, Patrick Mertens, and Dorit Merhof. Modeling the distribution of normal data in pre-trained deep features for anomaly detection. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 6726–6733. IEEE, 2021. 2

[37] Nicolae-Cătălin Ristea, Neelu Madan, Radu Tudor Ionescu, Kamal Nasrollahi, Fahad Shahbaz Khan, Thomas B Moeslund, and Mubarak Shah. Self-supervised predictive convolutional attentive block for anomaly detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13576–13586, 2022. 2

[38] Karsten Roth, Latha Pemula, Joaquin Zepeda, Bernhard Schölkopf, Thomas Brox, and Peter Gehler. Towards total recall in industrial anomaly detection. In *Proceedings of 2022 IEEE Conference on Computer Vision and Pattern Recognition*, pages 14298–14308, 2022. 2, 5, 6, 7, 8

[39] Marco Rudolph, Bastian Wandt, and Bodo Rosenhahn. Same same but differnet: Semi-supervised defect detection with normalizing flows. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 1907–1916, 2021. 2

[40] Mohammadreza Salehi, Niousha Sadjadi, Soroosh Baselizadeh, Mohammad H Rohban, and Hamid R Rabiee. Multiresolution knowledge distillation for anomaly detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14902–14912, 2021. 2

[41] Kihyuk Sohn, Chun-Liang Li, Jinsung Yoon, Minho Jin, and Tomas Pfister. Learning and evaluating representations for deep one-class classification. In *International Conference on Learning Representations*, 2021. 2

[42] Yu Sun, Xiaolong Wang, Zhuang Liu, John Miller, Alexei Efros, and Moritz Hardt. Test-time training with self-supervision for generalization under distribution shifts. In *International conference on machine learning*, pages 9229–9248. PMLR, 2020. 2, 3

[43] Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. In *International Conference on Learning Representations*, 2020. 2, 3

[44] Guodong Wang, Shumin Han, Errui Ding, and Di Huang. Student-teacher feature pyramid matching for anomaly detection. In *The British Machine Vision Conference (BMVC)*, 2021. 2

[45] Huajun Wang, Yuanhai Shao, Shenglong Zhou, Ce Zhang, and Naihua Xiu. Support vector machine classifier via $l_{0/1}$l0/1 soft-margin loss. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(10):7253–7265, 2022. 4

[46] Mei Wang and Weihong Deng. Deep visual domain adaptation: A survey. *Neurocomputing*, 312:135–153, 2018. 3

[47] Yue Wang, Jinlong Peng, Jiangning Zhang, Ran Yi, Yabiao Wang, and Chengjie Wang. Multimodal industrial anomaly detection via hybrid fusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8032–8041, 2023. 2, 5, 6, 7

[48] Julian Wyatt, Adam Leach, Sebastian M Schmon, and Chris G Willcocks. Anoddpm: Anomaly detection with denoising diffusion probabilistic models using simplex noise.

In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 650–656, 2022. 2

[49] Jie Yang, Yong Shi, and Zhiquan Qi. Dfr: Deep feature reconstruction for unsupervised anomaly segmentation. *arXiv preprint arXiv:2012.07122*, 2020. 2

[50] Minghui Yang, Peng Wu, and Hui Feng. Memseg: A semi-supervised method for image surface defect detection using differences and commonalities. *Engineering Applications of Artificial Intelligence*, 119:105835, 2023. 2

[51] Jihun Yi and Sungroh Yoon. Patch svdd: Patch-level svdd for anomaly detection and segmentation. In *Proceedings of the Asian conference on computer vision*, 2020. 2

[52] Seungdong Yoa, Seungjun Lee, Chiyoon Kim, and Hyunwoo J Kim. Self-supervised learning for anomaly detection with dynamic local augmentation. *IEEE Access*, 9:147201–147211, 2021. 2

[53] 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

[54] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In *BMVC*, 2016. 5, 6, 7, 8

[55] Vitjan Zavrtanik, Matej Kristan, and Danijel Skočaj. Draema discriminatively trained reconstruction embedding for surface anomaly detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8330–8339, 2021. 2

[56] Zheng Zhang and Xiaogang Deng. Anomaly detection using improved deep svdd model with data structure preservation. *Pattern Recognition Letters*, 148:1–6, 2021. 2

[57] Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip HS Torr, and Vladlen Koltun. Point transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16259–16268, 2021. 4, 5