AnomalyNCD: Towards Novel Anomaly Class Discovery in Industrial Scenarios

Supplementary Material

Overview

In this appendix, we provide additional descriptions of the following contents:

- Training details for novel class discovery (Sec. A).
- The implementation details about our experiment settings (Sec. B).
- The implementation details of our cropping method (Sec. C).
- Discussion of the anomaly-centered sub-image cropping (Sec. D).
- Discussion of the vision transformer under a supervised setting (Sec. E).
- Discussion of the labeled abnormal images (Sec. F).
- Discussion on handling the combined category (Sec. G).
- The computational analysis (Sec. H).
- The analysis of model performance using ground truth masks (Sec. I).
- Ablation study on the number of threshold index in MEBin (Sec. J).
- The sensitivity analysis of the hyperparameter τ in MEBin (Sec. K).
- Sensitivity analysis of threshold 0.5 in PLC (Sec. L).
- The binarization results of MEBin (Sec. M).
- More detailed t-SNE visualizations (Sec. N).
- More detailed quantitative results (Sec. O).
- Limitation (Sec. P).

A. Training details for novel class discovery

We present concrete implementations of the novel class discovery paradigm used in our method. Following [19, 20], we employ both classification learning and contrastive representation learning to train the network. For each sub-image $x_{i,k}$, we use a new subscript j to replace i, k for a simple and clear description. Thus the augmented views of x_j are formed as $(\tilde{x}_j, \hat{x}_j)$. These views are fed into MGViT, denoted as $f(\cdot)$, to extract the `[CLS]` tokens.

- For labeled sub-images, the ground truth labels are available. Therefore, we employ both self-supervised contrastive learning and supervised contrastive learning. In addition, supervised classification learning is performed by treating the ground truth labels as optimization targets.
- For unlabeled sub-images, we use self-supervised contrastive representation learning. The pseudo labels are generated to perform classification learning.

Contrastive representation learning. The `[CLS]` tokens extracted by MGViT are fed into a three-layer MLP, denoted as $\phi(\cdot)$, to generate the image features from $\tilde{x}_j$ and $\hat{x}_j$: $\tilde{\boldsymbol{z}}_j = \phi(f(\tilde{x}_j))$ and $\hat{\boldsymbol{z}}_j = \phi(f(\hat{x}_j))$. For both labeled and unlabeled sub-images, we employ the self-supervised contrastive loss $\mathcal{L}_{\text{rep}}$ as,

$$\mathcal{L}_{\text{rep}} = \frac{1}{|B|}\sum_{j\in B} -\log \frac{\exp(\hat{\boldsymbol{z}}_j^\top \tilde{\boldsymbol{z}}_j/\tau_u)}{\sum_{n\in\mathcal{N}_j}\exp(\hat{\boldsymbol{z}}_j^\top \tilde{\boldsymbol{z}}_n/\tau_u)} \tag{1}$$

where B denotes a mini-batch. $|B|$ represents the number of image pairs in B. $\mathcal{N}_j$ indexes other image pairs in the batch except $(\hat{x}_j, \tilde{x}_j)$, and τ_u is a temperature value. Similarly, the supervised contrastive loss for labeled data is written as:

$$\mathcal{L}_{\text{rep}}^{\mathbf{l}} = \frac{1}{|B^{\mathbf{l}}|}\sum_{j\in B^{\mathbf{l}}}\frac{1}{|\mathcal{P}_j|}\sum_{p\in\mathcal{P}_j} -\log \frac{\exp(\hat{\boldsymbol{z}}_j^\top \tilde{\boldsymbol{z}}_p/\tau_c)}{\sum_{n\in\mathcal{N}_j}\exp(\hat{\boldsymbol{z}}_j^\top \tilde{\boldsymbol{z}}_n/\tau_c)} \tag{2}$$

where $B^{\mathbf{l}} \subset B$ represents the indexes of labeled images in the batch B. $\mathcal{P}_j$ indexes other images in the batch B which have the same labels as $\hat{x}_j$ and $\tilde{x}_j$. τ_c is a temperature value.

Classification learning. For each image pair $(\hat{x}_j, \tilde{x}_j)$ in a batch, we feed them into the teacher and student networks respectively, which have shared MGViT and classification heads with different softmax temperatures. For labeled sub-images, we convert the ground truth label into the one-hot vector $\hat{y}_j = \tilde{y}_j \in \mathbb{R}^{1\times(\mathcal{C}^{\mathbf{l}}+\mathcal{C}^{\mathbf{u}})}$, where the entry corresponding to the target class is set to 1 and all other entries are 0. For unlabeled sub-images $\tilde{x}_j, \hat{x}_j$, we generate pseudo labels $\hat{q}_j, \tilde{q}_j$ by the "teacher" one, which employs a sharp temperature τ_t to produce confident predictions. Specifically, following DINO [6], we compute the logits for two views $(\hat{x}_j, \tilde{x}_j)$ as $\hat{l}_j = \mathcal{H}(f(\hat{x}_j))$ and $\tilde{l}_j = \mathcal{H}(f(\tilde{x}_j))$, where $\hat{l}_j, \tilde{l}_j \in \mathbb{R}^{1\times(\mathcal{C}^{\mathbf{l}}+\mathcal{C}^{\mathbf{u}})}$, $f(\cdot)$ indicates MGViT that outputs the `[CLS]` token, and $\mathcal{H}(\cdot)$ is the linear classifier. Then, a softmax with temperature τ_t converts these logits into pseudo labels $\hat{q}_j, \tilde{q}_j \in \mathbb{R}^{1\times(\mathcal{C}^{\mathbf{l}}+\mathcal{C}^{\mathbf{u}})}$ as follows,

$$\hat{q}_j^{(k)} = \frac{\exp(\hat{l}_j^{(k)}/\tau_t)}{\sum_{k=1}^{\mathcal{C}^{\mathbf{l}}+\mathcal{C}^{\mathbf{u}}}\exp(\hat{l}_j^{(k)}/\tau_t)} \tag{3}$$

where $\hat{q}_j^{(k)}$ and $\hat{l}_j^{(k)}$ denote the pseudo-label value and logit value respectively for the k-th class in $\mathcal{C}^{\mathbf{l}} + \mathcal{C}^{\mathbf{u}}$. Note that, we set the logits $\tilde{l}_j, \hat{l}_j$ for $\mathcal{C}^{\mathbf{l}}$ known classes to $-\infty$ to isolate known classes when calculating pseudo labels. Then the pseudo-label values for the known classes calculated by Eq. (3) are 0. The "student" one, using a smooth temperature τ_s, generates the probability predictions $\hat{p}_j \in \mathbb{R}^{1\times(\mathcal{C}^{\mathbf{l}}+\mathcal{C}^{\mathbf{u}})}$ for sub-image $\hat{x}_j$ as,

$$\hat{p}_j^{(k)} = \frac{\exp(\hat{l}_j^{(k)}/\tau_s)}{\sum_{k=1}^{\mathcal{C}^{\mathbf{l}}+\mathcal{C}^{\mathbf{u}}}\exp(\hat{l}_j^{(k)}/\tau_s)} \tag{4}$$

where $\hat{p}_j^{(k)}$ denotes the prediction probability for the k-th class in $\mathcal{C}^{\mathbf{l}} + \mathcal{C}^{\mathbf{u}}$. The student network is trained to align $\hat{p}_j$, $\tilde{p}_j$ with either the ground truth labels (for labeled data) or the pseudo labels (for unlabeled data). With the representation above, we leverage the standard cross-entropy loss $\mathcal{L}_{\text{CE}}(p, q) = -\sum_{c=0}^{\mathcal{C}^{\mathbf{l}}+\mathcal{C}^{\mathbf{u}}-1} p^c \log q^c$ to optimize the classification, which is represented as:

$$\mathcal{L}_{\text{cls}}^{\mathbf{l}} = \frac{1}{|B^{\mathbf{l}}|} \sum\nolimits_{j \in B^{\mathbf{l}}} (\mathcal{L}_{\text{CE}}(\hat{y}_j, \tilde{p}_j) + \mathcal{L}_{\text{CE}}(\tilde{y}_j, \hat{p}_j)) \quad (5)$$

$$\mathcal{L}_{\text{cls}}^{\mathbf{u}} = \frac{1}{|B^{\mathbf{u}}|} \sum\nolimits_{j \in B^{\mathbf{u}}} (\mathcal{L}_{\text{CE}}(\hat{q}_j, \tilde{p}_j) + \mathcal{L}_{\text{CE}}(\tilde{q}_j, \hat{p}_j)) \quad (6)$$

where $B^{\mathbf{u}}$ indexes the unlabeled images of the batch B.

Finally, we adopt a mean-entropy maximization regularizer [5] $\mathcal{L}_{\text{reg}}^{\mathbf{u}} = \mathcal{L}_{\text{CE}}(\bar{p}_j, \bar{p}_j)$ for unlabeled sub-images, where $\bar{p}_j = \frac{1}{2|B^{\mathbf{u}}|} \sum_{j \in B^{\mathbf{u}}} (\hat{p}_j + \tilde{p}_j)$ denotes the mean prediction of unlabeled images. The overall loss of training objective is written as:

$$\mathcal{L} = \lambda(\mathcal{L}_{\text{rep}}^{\mathbf{l}} + \mathcal{L}_{\text{cls}}^{\mathbf{l}}) + (1 - \lambda)(\mathcal{L}_{\text{rep}} + \mathcal{L}_{\text{cls}}^{\mathbf{u}} + \mu \mathcal{L}_{\text{reg}}^{\mathbf{u}}) \quad (7)$$

where λ is a hyperparameter that balances the supervised loss and the self-supervised loss. μ is the coefficient of the regularization.

B. Additional implementation details

We use ViT-B/8 [8] pre-trained with DINO [6] as our feature extractor. The self-attentions in the last 9 layers are replaced with our mask-guided attention. During training, all the layers of ViT are fixed except the last layer. As with UNO[9] and BYOP[21], we use the multi-head strategy in the classifier $\mathcal{H}$, and use the head with the smallest loss for inference. The input images are scaled to a resolution of 224×224. We set the batch size to 32 and epochs to 50. The Stochastic Gradient Descent (SGD) optimizer [4] is employed with a learning rate of 0.003. To generate two augmented views for contrastive learning, we follow the data augmentation strategies in BYOL [10] (random crop, flip, color jittering, and Gaussian blur) and RandAug [7] (rotation, posterize, and sharpness). Aligning with [20], the temperature values are set as follows: τ_u is 0.07 and τ_c is 1.0, τ_s is set to 0.1 and τ_t is initialized to 0.07. For the first 40 epochs, τ_t reduces every 4 epochs, linearly down to 0.04, then stays the same. For our Main Element Binarization approach, we set the $\mathcal{T}$ to 64 and τ to 4. In the overall loss, the parameters λ and μ are set to 0.3 and 4 respectively. In the Region Merging strategy, the temperature τ_α is set to 100 on the MVTec AD dataset and 50 on the MTD dataset.

For anomaly detection methods preceding AnomalyNCD, we choose zero-shot method MuSc [14], one-class methods PatchCore [16], EfficientAD [2], RD++ [18], PNI [1] and CPR [13]. Note that CPR does not provide the training code, we directly use the official checkpoints on the MVTec AD dataset. EfficientAD does not release official implementation, we use the unofficial version. For RD++ and PNI, we conduct experiments on the MTD dataset according to the configuration of MVTec AD.

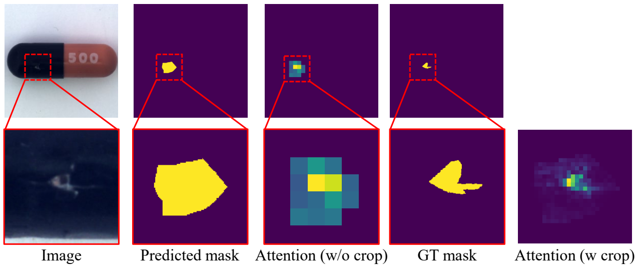


Figure 1. **Visualization of `[CLS]` tokens's attention** for entire image and sub-image.

C. Implementation details of cropping

For each region in $\mathbf{M}_i^{\mathbf{u}}$, we set a square box that completely encloses this region but has minimal area. Then we extend each square box by adding a 10% padding. The padding area includes the background of anomalies and thus enables our network to extract features indicating the anomaly's position on products, which is critical for classification. Notice that we keep the minimum crop size as 1% of the image size. Finally, along each square box of $\mathbf{M}_i^{\mathbf{u}}$, we crop a sub-image from $I_i^{\mathbf{u}}$ and its mask from $\mathbf{M}_i^{\mathbf{u}}$.

D. Discussion of the anomaly-centered sub-image cropping

The cropping operation plays a crucial role in industrial anomaly clustering for two primary reasons. First, many anomalies in industrial products are local and subtle. The cropping operation makes the anomalous region occupy most of the image, which helps the network learn the anomaly easily. Second, the cropping operation can handle images with *combined* type anomalies, where there are different types of anomalies in an image. More details about *combined* type anomalies are given in Sec. G.

As shown in Fig. 1, for the original image, the attention of `[CLS]` token has a coarse-grained response in the anomalous region. When the anomaly is subtle, the discriminative features are hard to learn. For the sub-image, the attention has a fine-grained response and learns the more discriminative pixels of subtle anomaly. We report multi-class classification on the MVTec AD and MTD datasets in Table 1. Since anomalies on the MTD dataset are subtle and finer, the crop operation is required for fine-grained feature learning, which can bring up to 23.8% NMI improvement for PatchCore+AnomalyNCD and 11.5% NMI gains for EfficientAD+AnomalyNCD. On the MVTec AD

Dataset	Setting	MuSc[14]+AnomalyNCD			PatchCore[16]+AnomalyNCD			EfficientAD[2]+AnomalyNCD			PNI[1]+AnomalyNCD		
		NMI	ARI	F_1	NMI	ARI	F_1	NMI	ARI	F_1	NMI	ARI	F_1
MVTec AD [3]	w/o crop	**0.622**	**0.549**	**0.740**	0.650	0.584	0.755	0.501	0.396	0.637	**0.676**	**0.616**	**0.777**
	w crop	0.613	0.526	0.712	**0.670**	**0.601**	**0.769**	**0.516**	**0.394**	**0.641**	0.675	0.609	0.769
MTD [11]	w/o crop	0.086	0.076	0.374	0.142	0.166	0.432	0.105	0.066	0.379	0.107	0.134	0.407
	w crop	**0.268**	**0.228**	**0.509**	**0.380**	**0.390**	**0.617**	**0.220**	**0.188**	**0.467**	**0.181**	**0.219**	**0.465**

Table 1. Ablation of Anomaly-Centered Sub-Image Cropping on the MVTec AD and MTD datasets.

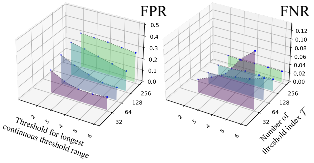


Figure 2. Ablation study on $\mathcal{T}$ in MEBin.

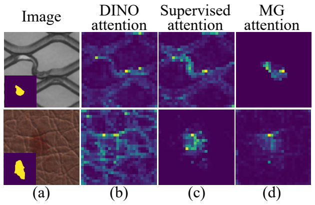


Figure 3. Visualization of self-attention of the [CLS] token.

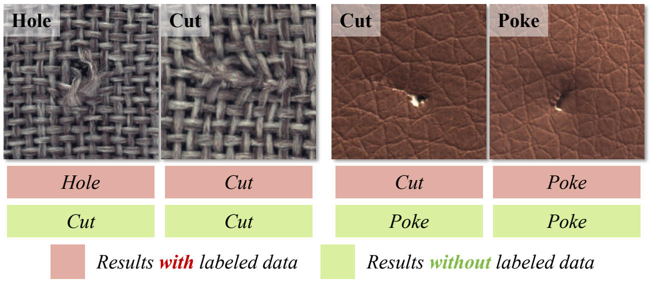


Figure 4. **Impact of using or not using labeled data**. We show two product cases, and each presents two similar anomalies.

dataset, the crop operation results in a 2.0% NMI increase for PatchCore+AnomalyNCD and 1.5% NMI gain for EfficientAD+AnomalyNCD. When using MuSc or PNI as the anomaly detection method, it only decreases NMI by 0.9% at most.

E. Discussion of ViT under a supervised setting

To study the ability of the vision transformer (ViT) to focus on local anomalies under a supervised setting, we leave one out cross validation. As shown in the attention maps of Fig. 3, the supervised training could focus on the anomaly regions, while it needs manually collecting and labeling the abnormal samples, which is time-consuming. As shown in (c) and (d), our AnomalyNCD leverages self-supervision to reduce these manual operations while also focuses attention on the anomalies, which is more suitable for industrial scenarios.

F. Discussion of the labeled abnormal images

As shown in Fig. 4, labeled data is significant for our mask-guided representation learning (MGRL). If only using unlabeled data, MGRL is prone to confuse anomalies with highly similar appearances. For example, a "Hole" on a carpet is treated as "Cut", and "Cut" on leather is misclassified as a "Poke". Actually, even humans tend to confuse them. With the supervision of labeled data, confusion does not occur. The reason is that the labeled data helps to build the feature space of prior classes, so that even the subtle differences from novel classes can be recognized.

	MVTec AD			MTD		
	NMI	ARI	F_1	NMI	ARI	F_1
w/o $\mathcal{D}^{\mathrm{l}}$	0.583	0.506	0.689	0.227	0.202	0.485
w $\mathcal{D}^{\mathrm{l}}$	**0.613**	**0.526**	**0.712**	**0.268**	**0.228**	**0.509**

Table 2. The ablation experiment of using labeled abnormal images $\mathcal{D}^{\mathrm{l}}$ on the MVTec AD and MTD datasets.

In the labeled abnormal images $\mathcal{D}^{\mathrm{l}}$, there is much prior knowledge of industrial anomalies classification, such as anomalies with similar size, color, and location belonging to the same class. Using them to train the network together can transfer the knowledge from $\mathcal{D}^{\mathrm{l}}$ to the network and separate confusing anomalies from each other in $\mathcal{D}^{\mathrm{u}}$. We report the quantitative results in Table 2, using $\mathcal{D}^{\mathrm{l}}$ brings 3.0% NMI improvements on MVTec AD and 3.9 % NMI improvements on MTD. In addition, we also use 15 categories in the MVTec AD dataset as the labeled abnormal images in turn. Except for the *toothbrush* category, which has only one anomaly class. The quantitative results are reported in Tables 3 to 5. MuSc is used as the anomaly detection method by default.

G. Discussion on handling the combined category

In main experiments, we follow Anomaly Clustering [17] to remove the combined class for a fair comparison. However, the combined class is quite common in the industrial scene, such as the *cable*, *pill*, *wood*, and *zipper* products on the MVTec AD dataset. The image with the combined class contains multiple different types of anomalies, yet current clustering methods can only classify the entire image into one anomaly type. For our method, we crop the image into many sub-images, and the classifier assigns each sub-image

train \ test	bottle	cable	capsule	carpet	grid	hazelnut	leather	metal_nut	pill	screw	tile	toothbrush	transistor	wood	zipper	mean
bottle	-	0.561	0.483	0.840	0.593	0.727	0.623	0.744	0.434	0.339	0.867	0.269	0.535	0.731	0.442	0.585
cable	0.676	-	0.522	0.862	0.699	0.695	0.784	0.799	0.396	0.435	0.858	0.341	0.511	0.640	0.545	0.626
capsule	0.614	0.604	-	0.714	0.652	0.715	0.724	0.674	0.412	0.433	0.719	0.242	0.498	0.702	0.586	0.603
carpet	0.564	0.571	0.503	-	0.624	0.714	0.818	0.745	0.416	0.418	0.885	0.218	0.508	0.681	0.541	0.586
grid	0.552	0.520	0.460	0.790	-	0.747	0.714	0.707	0.441	0.361	0.794	0.218	0.517	0.680	0.555	0.575
hazelnut	0.535	0.616	0.451	0.731	0.660	-	0.640	0.799	0.401	0.351	0.824	0.299	0.513	0.712	0.493	0.573
leather	0.639	0.603	0.443	0.755	0.597	0.740	-	0.658	0.439	0.419	0.885	0.368	0.507	0.663	0.570	0.592
metal_nut	0.598	0.556	0.575	0.863	0.723	0.684	0.838	-	0.463	0.355	0.811	0.242	0.521	0.617	0.559	0.600
pill	0.603	0.594	0.430	0.850	0.641	0.753	0.763	0.728	-	0.398	0.885	0.368	0.543	0.678	0.518	0.625
screw	0.569	0.559	0.486	0.746	0.501	0.688	0.815	0.787	0.433	-	0.895	0.368	0.493	0.685	0.542	0.612
tile	0.514	0.587	0.506	0.762	0.501	0.750	0.790	0.725	0.425	0.414	-	0.242	0.483	0.693	0.548	0.565
transistor	0.499	0.607	0.420	0.645	0.569	0.699	0.651	0.768	0.446	0.387	0.738	0.299	-	0.445	0.553	0.552
wood	0.586	0.633	0.472	0.764	0.655	0.708	0.700	0.789	0.402	0.354	0.767	0.218	0.482	-	0.533	0.576
zipper	0.627	0.555	0.395	0.795	0.616	0.723	0.765	0.755	0.432	0.412	0.803	0.398	0.543	0.658	-	0.606
mean	0.583	0.582	0.473	0.778	0.618	0.719	0.740	0.744	0.426	0.390	0.825	0.292	0.512	0.660	0.537	0.591

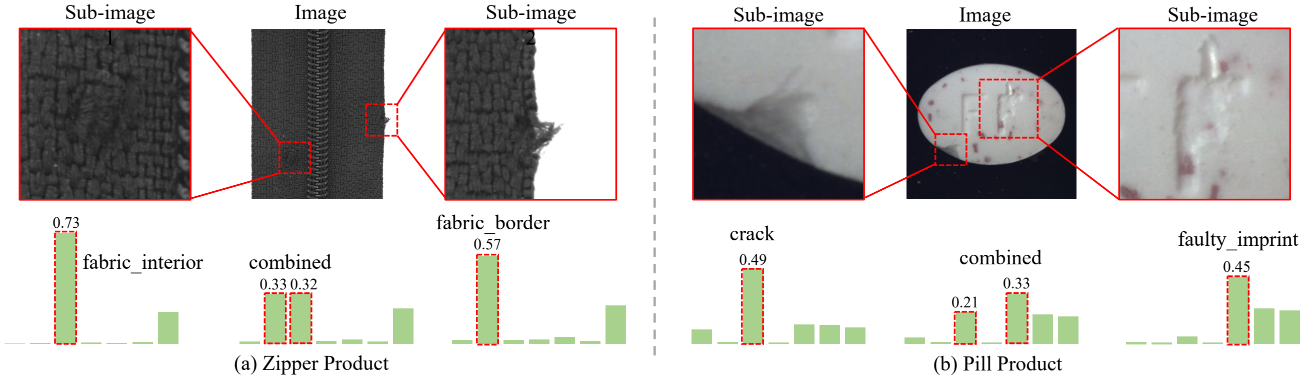
Table 3. The detailed NMI metric evaluated on different labeled abnormal images $\mathcal{D}^1$ on the MVTec AD dataset.

train \ test	bottle	cable	capsule	carpet	grid	hazelnut	leather	metal_nut	pill	screw	tile	toothbrush	transistor	wood	zipper	mean
bottle	-	0.608	0.402	0.778	0.457	0.704	0.569	0.623	0.325	0.242	0.816	0.259	0.442	0.660	0.322	0.515
cable	0.643	-	0.384	0.783	0.625	0.712	0.689	0.659	0.267	0.297	0.835	0.210	0.474	0.478	0.404	0.533
capsule	0.554	0.628	-	0.600	0.585	0.705	0.648	0.544	0.302	0.320	0.575	0.211	0.411	0.633	0.445	0.517
carpet	0.495	0.602	0.352	-	0.546	0.715	0.789	0.606	0.289	0.315	0.830	0.167	0.460	0.599	0.409	0.512
grid	0.532	0.580	0.313	0.696	-	0.744	0.672	0.563	0.318	0.257	0.744	0.167	0.471	0.555	0.430	0.503
hazelnut	0.474	0.643	0.361	0.679	0.610	-	0.581	0.659	0.262	0.257	0.781	0.312	0.430	0.617	0.378	0.503
leather	0.586	0.643	0.295	0.688	0.533	0.717	-	0.515	0.298	0.310	0.830	0.259	0.407	0.569	0.416	0.505
metal_nut	0.562	0.482	0.452	0.789	0.664	0.702	0.765	-	0.313	0.301	0.773	0.211	0.428	0.486	0.422	0.525
pill	0.562	0.501	0.303	0.791	0.549	0.763	0.671	0.573	-	0.273	0.830	0.259	0.464	0.608	0.395	0.539
screw	0.519	0.618	0.334	0.707	0.394	0.707	0.743	0.675	0.286	-	0.849	0.259	0.416	0.635	0.437	0.541
tile	0.478	0.635	0.367	0.697	0.420	0.745	0.679	0.570	0.278	0.322	-	0.211	0.380	0.570	0.429	0.477
transistor	0.467	0.544	0.313	0.551	0.496	0.700	0.588	0.640	0.296	0.301	0.670	0.312	-	0.347	0.401	0.473
wood	0.539	0.639	0.370	0.684	0.599	0.719	0.620	0.656	0.285	0.232	0.732	0.167	0.416	-	0.395	0.504
zipper	0.571	0.476	0.268	0.769	0.566	0.740	0.726	0.629	0.274	0.286	0.747	0.313	0.455	0.585	-	0.529
mean	0.537	0.585	0.347	0.709	0.542	0.721	0.672	0.609	0.292	0.286	0.770	0.237	0.435	0.565	0.406	0.513

Table 4. The detailed ARI metric evaluated on different labeled abnormal images $\mathcal{D}^1$ on the MVTec AD dataset.

train \ test	bottle	cable	capsule	carpet	grid	hazelnut	leather	metal_nut	pill	screw	tile	toothbrush	transistor	wood	zipper	mean
bottle	-	0.655	0.636	0.821	0.641	0.809	0.661	0.678	0.500	0.450	0.923	0.762	0.640	0.853	0.489	0.680
cable	0.843	-	0.636	0.812	0.795	0.827	0.766	0.687	0.460	0.513	0.932	0.738	0.620	0.662	0.600	0.706
capsule	0.795	0.676	-	0.709	0.718	0.782	0.694	0.617	0.480	0.513	0.740	0.738	0.630	0.838	0.681	0.702
carpet	0.747	0.640	0.644	-	0.705	0.782	0.847	0.661	0.453	0.506	0.932	0.714	0.630	0.824	0.607	0.692
grid	0.795	0.640	0.614	0.778	-	0.818	0.774	0.617	0.507	0.456	0.855	0.714	0.660	0.794	0.652	0.691
hazelnut	0.699	0.705	0.636	0.778	0.782	-	0.694	0.687	0.453	0.475	0.889	0.786	0.620	0.829	0.533	0.683
leather	0.807	0.676	0.553	0.795	0.679	0.791	-	0.617	0.487	0.525	0.932	0.762	0.620	0.809	0.644	0.693
metal_nut	0.807	0.619	0.652	0.803	0.833	0.800	0.798	-	0.533	0.513	0.880	0.738	0.630	0.647	0.622	0.705
pill	0.807	0.626	0.576	0.838	0.718	0.864	0.742	0.643	-	0.488	0.932	0.762	0.620	0.838	0.600	0.718
screw	0.783	0.647	0.652	0.795	0.577	0.773	0.774	0.704	0.507	-	0.940	0.762	0.620	0.838	0.652	0.716
tile	0.759	0.691	0.667	0.735	0.564	0.818	0.750	0.617	0.447	0.506	-	0.738	0.600	0.794	0.645	0.665
transistor	0.675	0.647	0.591	0.658	0.667	0.800	0.702	0.678	0.513	0.506	0.786	0.786	-	0.618	0.637	0.662
wood	0.795	0.683	0.629	0.761	0.769	0.800	0.702	0.687	0.493	0.444	0.855	0.714	0.630	-	0.607	0.684
zipper	0.807	0.633	0.545	0.846	0.705	0.836	0.823	0.687	0.467	0.494	0.880	0.786	0.630	0.824	-	0.712
mean	0.778	0.657	0.618	0.779	0.704	0.808	0.748	0.660	0.485	0.491	0.883	0.750	0.627	0.782	0.613	0.694

Table 5. The detailed F_1 metric evaluated on different labeled abnormal images $\mathcal{D}^1$ on the MVTec AD dataset.



(a) Zipper Product

(b) Pill Product

Figure 5. **Multi-class classification results of two combined-type anomaly images.** Taking "zipper" and "pill" as examples, we show the predicted probabilities for each sub-image individually and for the entire image.

	(II)	(III)	(IV)	Total
Time (ms)	129.8	22.8	0.5	153.1

Table 6. Inference time for each module on a per-image basis.

a label. In this way, all the anomalous regions in the combined image can be found and classified individually. As shown in Fig. 5, there are two types of anomalies in the zipper and pill products respectively. Each anomalous region is cropped and correctly classified. By merging the prediction of the sub-images to the entire image, we find that classifying the image as these two types of anomalies has a higher probability.

H. Computational analysis

We measure the inference speed of AnomalyNCD on an NVIDIA RTX 3090 GPU. The entire inference process consists of four steps: (I) applying an anomaly detection method to the input image, (II) binarizing the anomaly map and cropping sub-images accordingly, (III) feeding these sub-images and their corresponding masks into our mask-guided ViT and getting probability output, and (IV) merging predictions from each sub-image to generate the final output. Table 6 presents the per-image inference time for each step. Note that the time taken for step (I) depends on the anomaly detection method and is independent of our AnomalyNCD. The majority of the inference time is concentrated in the MEBin and image cropping, which account for over 80% of the total runtime. MEBin uses the official OpenCV library for connection component calculations, resulting in the inability to use CUDA acceleration.

I. Analysis of model performance using ground truth masks

In the experimental section, we demonstrate that our multi-class classification performs better when the anomaly detection method performs better. So in order to test the optimal results of our AnomalyNCD, we assume an ideal anomaly detection method where the ground truth masks of unlabeled images are available. We report the results in Table 7, our approach achieves significant improvement over Anomaly Clustering on two datasets. On MVTec AD, there is a 16.0% improvement on the NMI and a 19.1% improvement on the F_1, and on MTD, there is a 36.2% improvement on the NMI and a 35.9% improvement on the F_1. In this ideal case, ground truth masks do not introduce any over-detections and missed detections compared to anomaly maps generated by anomaly detection methods. With the improvement of anomaly detection methods in the future, our AnomalyNCD can achieve better results.

	MVTec AD			MTD		
Methods	NMI	ARI	F_1	NMI	ARI	F_1
AC [17]	0.711	0.638	0.718	0.467	0.359	0.482
Ours	**0.871**	**0.851**	**0.909**	**0.829**	**0.863**	**0.841**

Table 7. The quantitative results of using the ground truth masks. We compare our AnomalyNCD with Anomaly Clustering on the MVTec AD and MTD datasets.

J. Ablation study on the number of threshold index in MEBin

In Fig. 2, we conduct the ablation study on the number of threshold index $\mathcal{T}$. Note that we use powers of 2 for candidates of $\mathcal{T}$, since it is calculated on grayscale images (0-255). $\mathcal{T}=64$ achieves better FPR-FNR trade-off compared to other $\mathcal{T}$. With a small value ($\mathcal{T}=32$), the binary mask changes dramatically, making it difficult to determine a stable connected component. Conversely, a larger $\mathcal{T}$ brings more time consumption. Therefore, we set $\mathcal{T}=64$ as the default value.

PLC Thresholds	MVTec AD - MuSc	MTD - MuSc	MVTec AD - RD++	MTD - RD++	MVTec AD - PatchCore	MTD - PatchCore	Rank
0.1	0.602/0.522/0.715	0.212/0.180/0.419	0.625/0.531/0.717	0.368/0.377/0.587	0.630/0.574/0.734	0.457/**0.453**/0.683	3.44
0.3	0.615/0.533/0.727	0.230/0.204/0.487	0.627/0.545/0.721	**0.391/0.382**/0.595	0.646/0.594/0.757	0.466/0.452/0.683	2.28
0.5	0.613/0.526/0.712	**0.268/0.228/0.509**	0.631/0.542/0.721	0.368/0.361/**0.600**	**0.670/0.601/0.769**	0.380/0.390/**0.715**	**2.17**
0.7	**0.630/0.557/0.731**	0.257/0.212/0.489	**0.650/0.561/0.741**	0.353/0.333/0.566	0.653/0.578/0.748	**0.602**/0.415/0.664	2.22
0.9	0.584/0.476/0.682	0.231/0.172/0.489	0.604/0.487/0.682	0.337/0.302/0.549	0.506/0.338/0.610	0.423/0.352/0.660	4.67

Table 8. The results (**NMI**, **ARI**, **F1**) on two datasets, when taking various PLC thresholds. **Rank** is the average ranking of the threshold.

	MuSc[14]+AnomalyNCD			CPR[13]+AnomalyNCD		
	NMI	ARI	F_1	NMI	ARI	F_1
$\tau = 2$	0.640	0.537	0.707	0.720	0.647	0.787
$\tau = 3$	0.647	0.543	0.718	0.736	0.668	0.797
$\tau = 4$	0.613	0.526	0.712	0.736	0.674	0.805
$\tau = 5$	0.618	0.539	0.740	0.721	0.665	0.807
$\tau = 6$	0.600	0.510	0.715	0.710	0.655	0.799
variance	0.0003	0.0001	0.0001	0.0001	0.0001	0.0001

Table 9. Sensitivity analysis of the minimum stable range τ in MEBin. We use the zero-shot AD method MuSc and one-class AD method CPR to conduct experiments.

K. Sensitivity analysis of τ in MEBin

In Table 9, we experiment with the parameter sensitivity of the minimum stable range τ on the MVTec AD dataset. We use the zero-shot anomaly detection method MuSc [14] and the one-class method CPR [13] respectively to combine with our AnomalyNCD. When the hyperparameter τ is changed, the multi-class anomaly classification results change with a variance of about 0.0001, demonstrating that our method is insensitive to τ.

L. Sensitivity analysis of threshold 0.5 in PLC

Although MEBin achieves a decent trade-off, it may still get over-detections due to local noises. Thus, PLC uses the threshold 0.5 to find over-detections and corrects their pseudo label with a normal one-hot label. The optimal PLC threshold is related to datasets and anomaly detection methods. Tab. 8 shows the results of three methods on two datasets, when taking various PLC thresholds. Observably, a threshold of 0.5 ranks 1st performance on average in all candidates.

M. Binarization results of MEBin

From Fig. 9 to Fig. 11, we binarize the anomaly maps output by MuSc [14] using different binarization strategies, including the fixed threshold, the Otsu [15] method, and our MEBin. Our binarization strategy does not require the validation set to determine a threshold and adaptively obtain the optimal threshold for each image with fewer false positives and false negatives. Compared to the Otsu method, our MEBin has fewer false positives, especially on normal images. We also report the optimal threshold searched by our MEBin in the last column of Fig. 9 to Fig. 11. The optimal threshold ranges from 0.380 to 1.000, making it difficult to handle all situations with a fixed threshold.

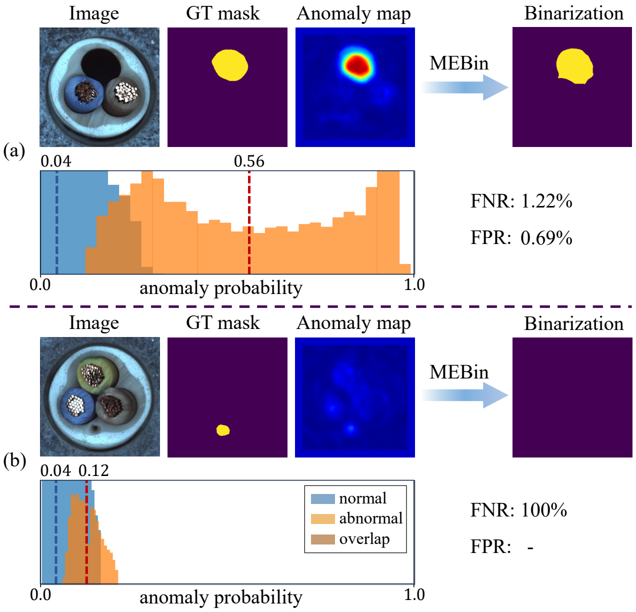


Figure 6. **Analysis of anomaly maps in EfficientAD.** We visualize the pixel-level anomaly probability histogram, and use dashed lines to represent the average anomaly probability for normal and anomalous regions respectively.

N. Detailed t-SNE visualization

In Fig. 7, we show more t-SNE visualization on all the products of the MVTec AD dataset. We also provide more qualitative comparisons with AC [17] and Uniformaly [12] in Fig. 8. These methods show slight clustering phenomena, while our AnomalyNCD has larger inter-class distances and smaller intra-class distances. This further proves the superiority of our method from the qualitative perspective.

O. Detailed quantitative results

In this section, we report the detailed results of our AnomalyNCD combined with various anomaly detection methods on the MVTec AD dataset, as shown in Table 10.

P. Limitation

The anomaly detection methods used affect the performance of our AnomalyNCD. In general, if the anomaly detection method has a higher AUPRO, we can achieve a better performance in the multi-class anomaly classification task. In the ideal situation where the ground truth mask is available (AUPRO=100%), our AnomalyNCD is significantly superior to other methods, which has been discussed in Sec. I. However, EfficientAD [2] is an exception, with a lower multi-class anomaly classification performance despite higher AUPRO. We observe that a large span of anomaly probability values exists between different anomaly maps generated by EfficientAD, as shown in Fig. 6. The average value of the anomalous region in (a) is only 0.12, which is close to the anomaly probability in the normal region, while that in (b) is 0.56. This large span causes false negatives in (a) during binarization, resulting in a lower performance for anomaly classification.

Methods	**MuSc[14] +AnomalyNCD**			**PatchCore[16] +AnomalyNCD**			**EfficientAD[2] +AnomalyNCD**			**RD++[18] +AnomalyNCD**			**PNI[1] +AnomalyNCD**			**CPR[13] +AnomalyNCD**		
Products\Metric	NMI	ARI	F_1	NMI	ARI	F_1	NMI	ARI	F_1	NMI	ARI	F_1	NMI	ARI	F_1	NMI	ARI	F_1
bottle	0.613	0.583	0.819	0.734	0.671	0.855	0.544	0.467	0.759	0.640	0.529	0.771	0.676	0.598	0.819	0.775	0.757	0.904
cable	0.597	0.492	0.626	0.758	0.666	0.741	0.549	0.414	0.619	0.679	0.602	0.734	0.765	0.601	0.691	0.711	0.476	0.561
capsule	0.445	0.335	0.591	0.402	0.277	0.530	0.519	0.402	0.644	0.481	0.392	0.606	0.425	0.338	0.568	0.549	0.438	0.674
carpet	0.852	0.837	0.906	0.628	0.569	0.726	0.672	0.577	0.726	0.599	0.527	0.658	0.619	0.548	0.701	0.740	0.638	0.795
grid	0.622	0.578	0.731	0.670	0.581	0.731	0.684	0.570	0.795	0.631	0.544	0.615	0.583	0.534	0.705	0.766	0.689	0.731
hazelnut	0.662	0.582	0.718	0.859	0.875	0.927	0.570	0.429	0.673	0.845	0.853	0.909	0.919	0.926	0.955	0.723	0.727	0.827
leather	0.863	0.838	0.911	0.770	0.734	0.839	0.587	0.449	0.621	0.746	0.688	0.782	0.672	0.616	0.734	0.865	0.827	0.895
metal_nut	0.643	0.467	0.565	0.910	0.883	0.948	0.436	0.272	0.443	0.851	0.821	0.922	0.891	0.872	0.948	0.870	0.848	0.930
pill	0.439	0.291	0.513	0.476	0.335	0.567	0.388	0.194	0.460	0.467	0.286	0.527	0.461	0.315	0.580	0.694	0.592	0.747
screw	0.399	0.265	0.488	0.638	0.569	0.756	0.594	0.498	0.719	0.615	0.551	0.725	0.640	0.616	0.769	0.742	0.698	0.800
tile	0.885	0.850	0.940	0.927	0.912	0.966	0.523	0.375	0.598	0.757	0.711	0.795	1.000	1.000	1.000	0.98	0.976	0.992
toothbrush	0.368	0.259	0.762	0.271	0.084	0.667	0.398	0.313	0.786	0.316	0.164	0.714	0.271	0.084	0.667	0.508	0.499	0.857
transistor	0.531	0.421	0.620	0.642	0.693	0.760	0.439	0.462	0.650	0.477	0.292	0.500	0.716	0.743	0.820	0.577	0.523	0.700
wood	0.743	0.672	0.868	0.782	0.644	0.868	0.375	0.196	0.574	0.750	0.651	0.868	0.850	0.790	0.927	0.816	0.756	0.912
zipper	0.526	0.417	0.615	0.583	0.518	0.652	0.455	0.291	0.548	0.608	0.515	0.689	0.639	0.552	0.659	0.726	0.667	0.756
Mean	0.613	0.526	0.712	0.670	0.601	0.769	0.516	0.394	0.641	0.631	0.542	0.721	0.675	0.609	0.769	0.736	0.674	0.805

Table 10. Detailed quantitative results on the MVTec AD dataset.

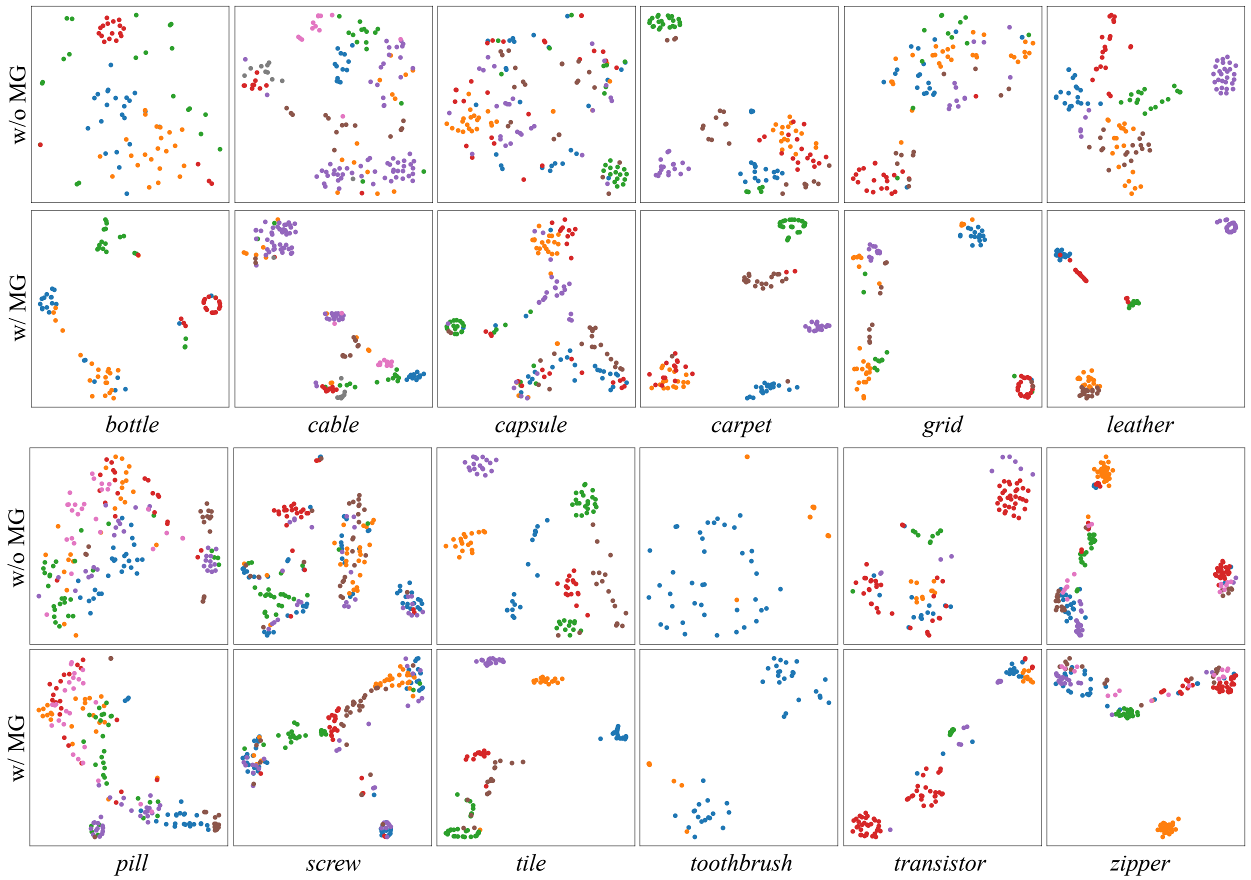


Figure 7. **T-SNE visualization of sub-images on the MVTec AD dataset.** The different colors of dots represent their anomaly classes.

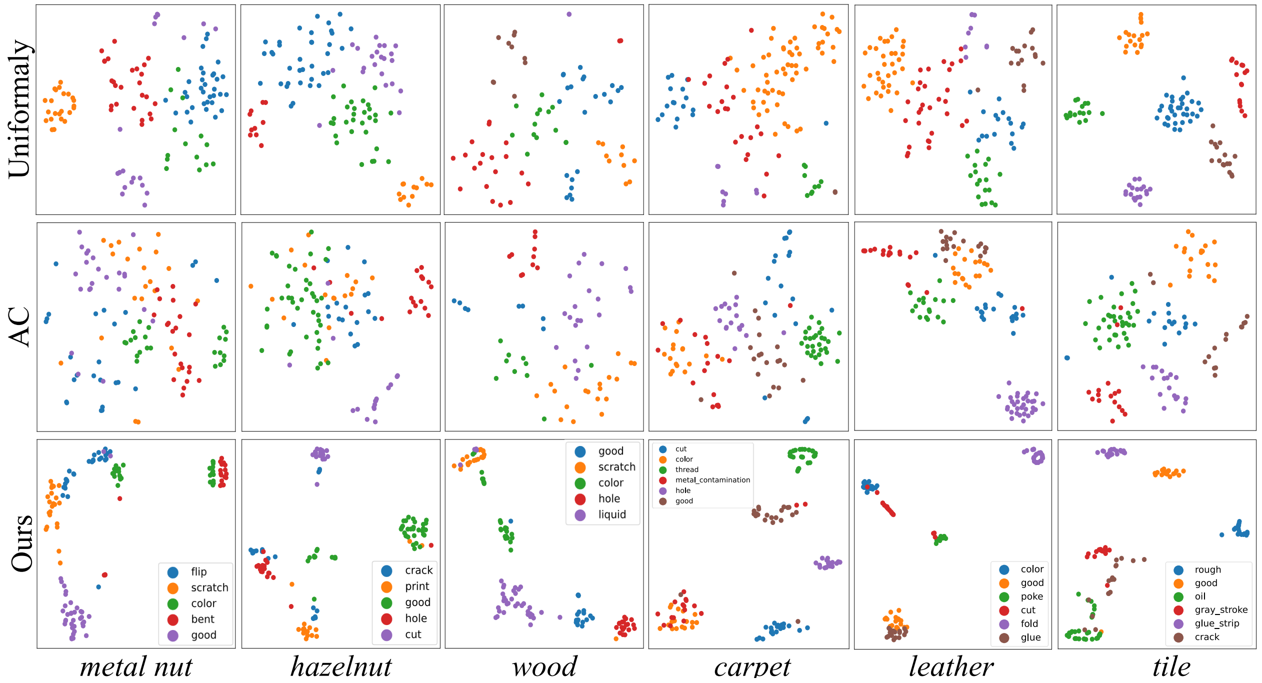


Figure 8. Qualitative comparisons with different clustering methods by t-SNE visualization.

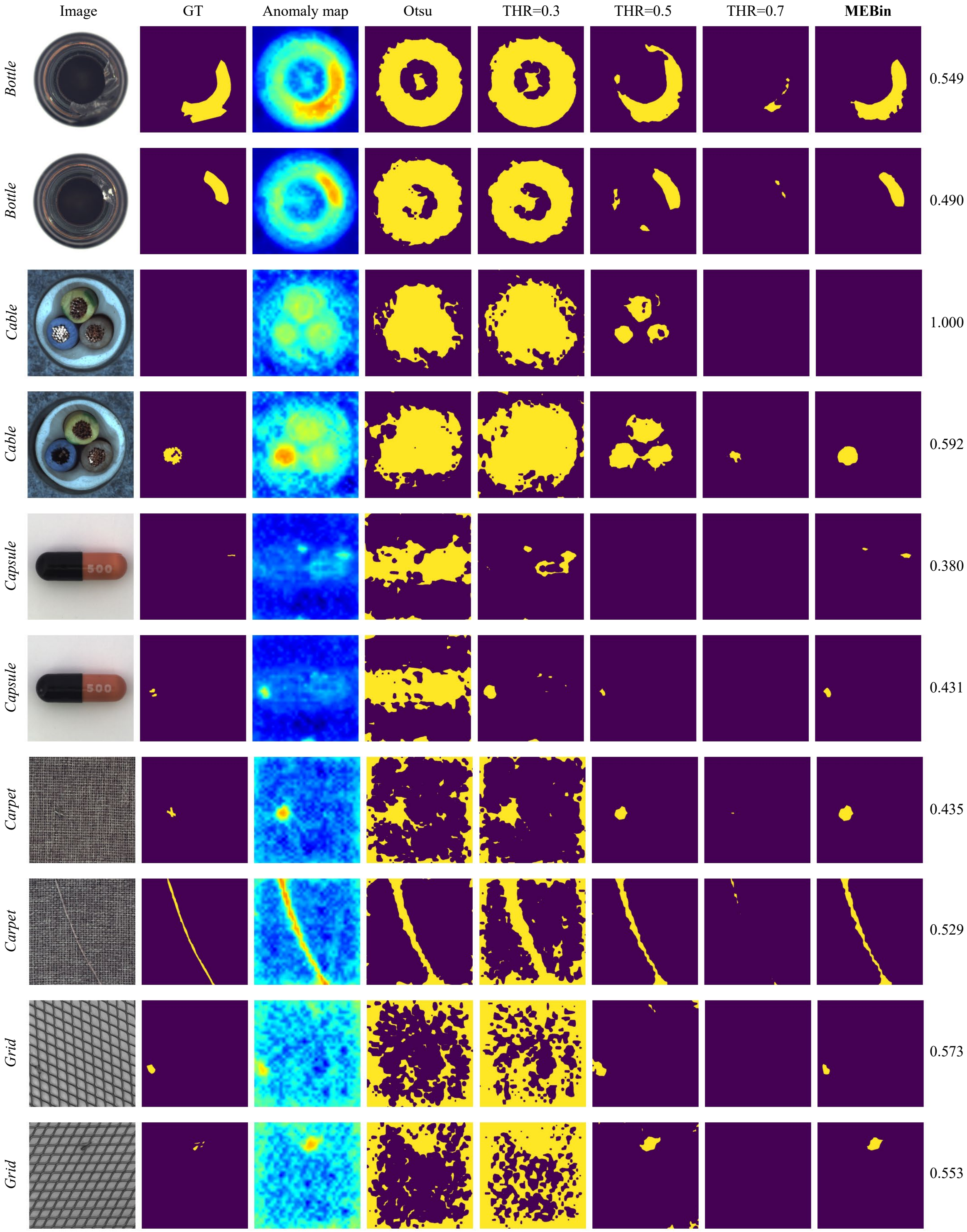


Figure 9. **Binarization results of MEBin on the MVTec AD dataset.** We show for each category: RGB image, ground truth, anomaly map, binary map of Otsu method, binary map of fixed threshold 0.3, binary map of fixed threshold 0.5, binary map of fixed threshold 0.7, and binary map of our MEBin. We report the optimal threshold searched by our MEBin in the last column.

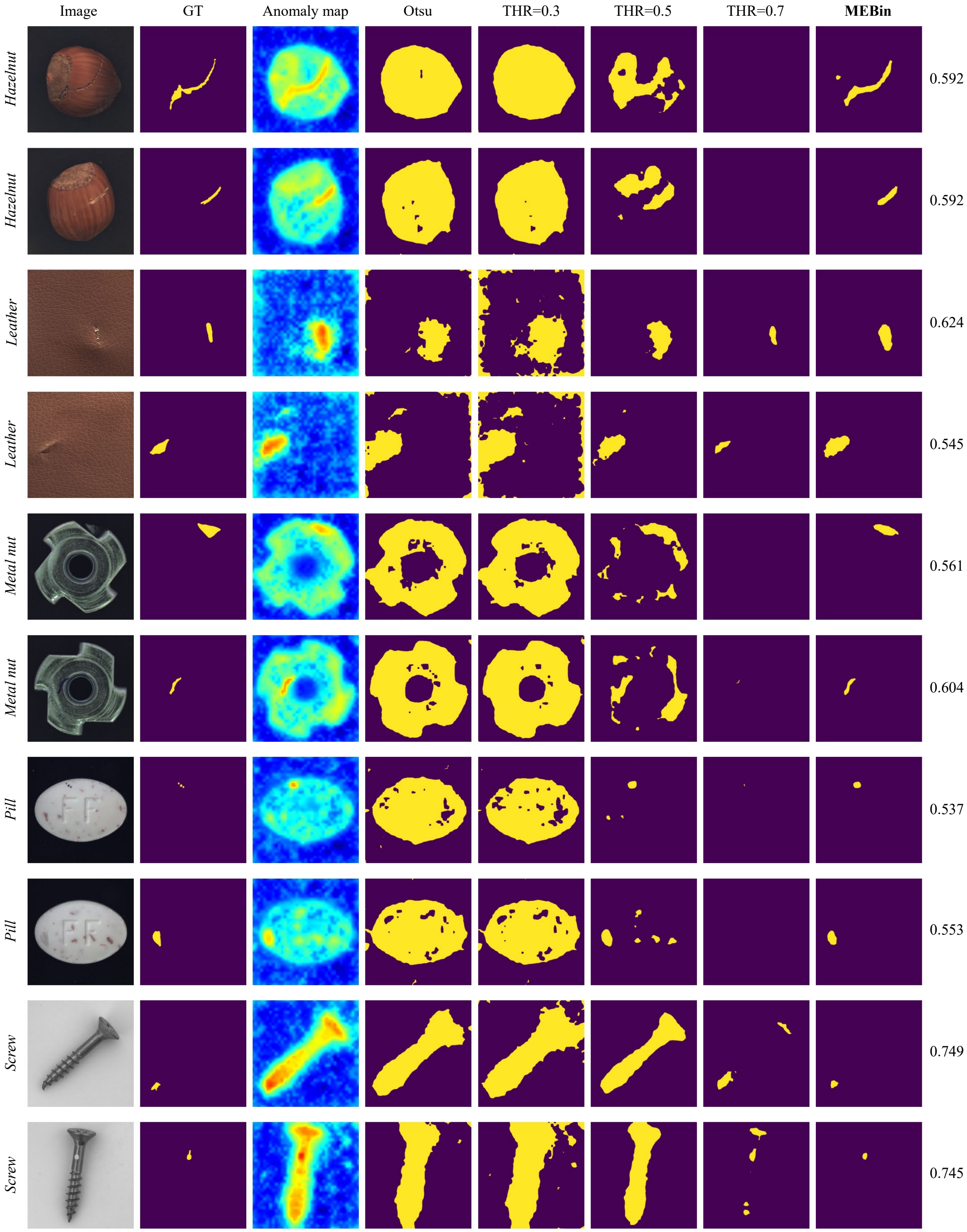


Figure 10. **Binarization results of MEBin on the MVTec AD dataset.** We show for each category: RGB image, ground truth, anomaly map, binary map of Otsu method, binary map of fixed threshold 0.3, binary map of fixed threshold 0.5, binary map of fixed threshold 0.7, and binary map of our MEBin. We report the optimal threshold searched by our MEBin in the last column.

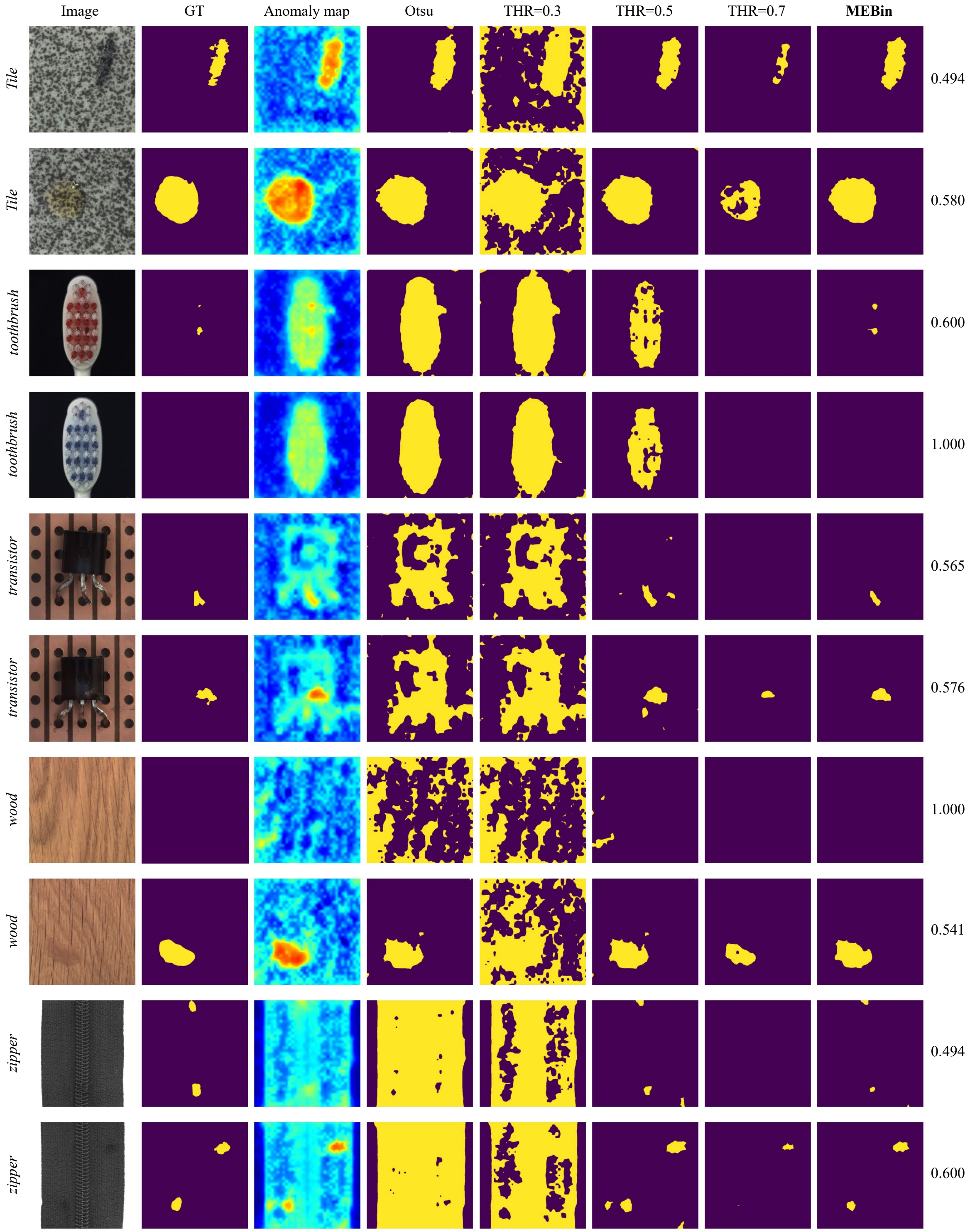


Figure 11. **Binarization results of MEBin on the MVTec AD dataset.** We show for each category: RGB image, ground truth, anomaly map, binary map of Otsu method, binary map of fixed threshold 0.3, binary map of fixed threshold 0.5, binary map of fixed threshold 0.7, and binary map of our MEBin. We report the optimal threshold searched by our MEBin in the last column.

References

[1] Jaehyeok Bae, Jae-Han Lee, and Seyun Kim. Pni: industrial anomaly detection using position and neighborhood information. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023. 2, 3, 7

[2] Kilian Batzner, Lars Heckler, and Rebecca König. Efficientad: Accurate visual anomaly detection at millisecond-level latencies. In *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2024. 2, 3, 7

[3] Paul Bergmann, Michael Fauser, David Sattlegger, and Carsten Steger. Mvtec ad–a comprehensive real-world dataset for unsupervised anomaly detection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 3

[4] Léon Bottou. Large-scale machine learning with stochastic gradient descent. In *International Conference on Computational Statistics (COMPSTAT)*, 2010. 2

[5] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. 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 *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. 1, 2

[7] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshop (CVPR Workshop)*, 2020. 2

[8] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations (ICLR)*, 2021. 2

[9] Enrico Fini, Enver Sangineto, Stéphane Lathuiliere, Zhun Zhong, Moin Nabi, and Elisa Ricci. A unified objective for novel class discovery. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. 2

[10] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. 2

[11] Yibin Huang, Congying Qiu, and Kui Yuan. Surface defect saliency of magnetic tile. *The Visual Computer*, 36(1):85–96, 2020. 3

[12] Yujin Lee, Harin Lim, Seoyoon Jang, and Hyunsoo Yoon. Uniformaly: Towards task-agnostic unified framework for visual anomaly detection. *arXiv preprint arXiv:2307.12540*, 2023. 6

[13] Hanxi Li, Jianfei Hu, Bo Li, Hao Chen, Yongbin Zheng, and Chunhua Shen. Target before shooting: Accurate anomaly detection and localization under one millisecond via cascade patch retrieval. *IEEE Transactions on Image Processing (TIP)*, 2023. 2, 6, 7

[14] Xurui Li, Ziming Huang, Feng Xue, and Yu Zhou. Musc: Zero-shot industrial anomaly classification and segmentation with mutual scoring of the unlabeled images. In *International Conference on Learning Representations (ICLR)*, 2024. 2, 3, 6, 7

[15] Nobuyuki Otsu. A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics (TSMC)*, 9(1):62–66, 1979. 6

[16] Karsten Roth, Latha Pemula, Joaquin Zepeda, Bernhard Schölkopf, Thomas Brox, and Peter Gehler. Towards total recall in industrial anomaly detection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 2, 3, 7

[17] Kihyuk Sohn, Jinsung Yoon, Chun-Liang Li, Chen-Yu Lee, and Tomas Pfister. Anomaly clustering: Grouping images into coherent clusters of anomaly types. In *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2023. 3, 5, 6

[18] Tran Dinh Tien, Anh Tuan Nguyen, Nguyen Hoang Tran, Ta Duc Huy, Soan Duong, Chanh D Tr Nguyen, and Steven QH Truong. Revisiting reverse distillation for anomaly detection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 2, 7

[19] Sagar Vaze, Kai Han, Andrea Vedaldi, and Andrew Zisserman. Generalized category discovery. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 1

[20] Xin Wen, Bingchen Zhao, and Xiaojuan Qi. Parametric classification for generalized category discovery: A baseline study. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023. 1, 2

[21] Muli Yang, Liancheng Wang, Cheng Deng, and Hanwang Zhang. Bootstrap your own prior: Towards distribution-agnostic novel class discovery. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 2