Multimodal Industrial Anomaly Detection by Crossmodal Feature Mapping Supplementary Material

Alex Costanzino* Pierluigi Zama Ramirez* Giuseppe Lisanti Luigi Di Stefano

CVLAB, Department of Computer Science and Engineering (DISI) – University of Bologna, Italy

https://cvlab-unibo.github.io/CrossmodalFeatureMapping/

Overview

This supplementary material includes additional experimental results. In particular, we report:

- A more detailed analysis on the dynamic of the PRO (Per-Region Overlap) curve, alongside comparisons dealing with different integration thresholds;
- An ablation study concerning the architecture of the Feature Mapping networks, *i.e*. the core components in our method;
- An ablation study regarding the backbone employed as 2D Feature Extractor;
- Additional quantitative and qualitative results dealing with both MVTec 3D-AD and Eyecandies.

A. Analysis of the PRO curve

The chart in Fig. 1 reports the Per-Region Overlap curve provided by our method on class *Foam* of the MVTec 3D-AD dataset. The chart shows how most of the dynamic of the curve is concentrated way underneath the 0.3 integration threshold used to define the popular AUPRO@30% metric. This is also highlighted in Fig. 2, which compares the different Multimodal AD methods focusing on lower FPRs.

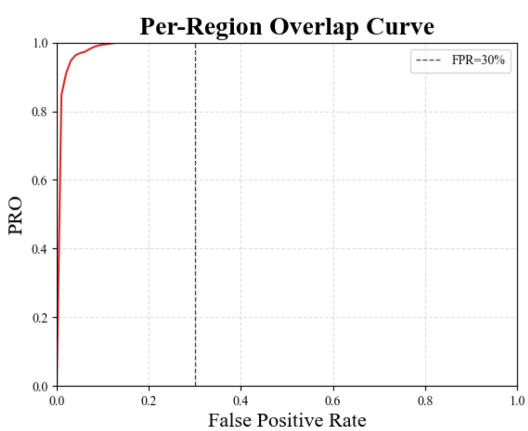

Figure 1. **PRO curve - Whole FPR Range.** Per-Region Overlap curve obtained by our method on class *Foam* of MVTec 3D-AD. The dotted line shows the AUPRO@30% threshold.

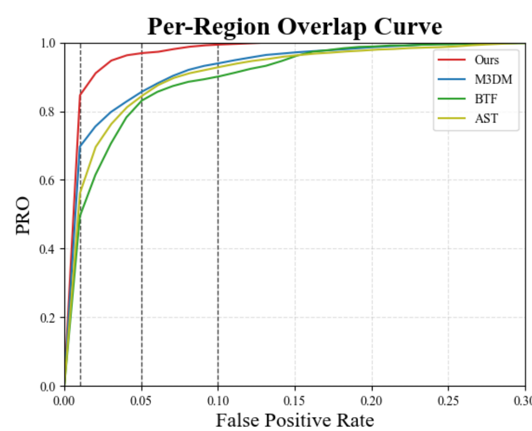

Figure 2. **PRO curve - Lower FPRs.** Per-Region Overlap curve obtained by all Multmodal AD methods on class *Foam* of MVTec 3D-AD. Focus on the [0-0.3] FPR range.

Thus, as discussed in the main paper, on one hand choosing FPR=0.3 as integration threshold may not match the requirements of a number of industrial applications, on the other, it tends to wash out the performance differences between the methods, which, indeed, behave much more differently at lower, *i.e*., more challenging FPRs. Hence, we deem it worth considering also more demanding variants of the AUPRO metric, such as, in particular, those obtained with integration thresholds

*These authors contributed equally to this work.

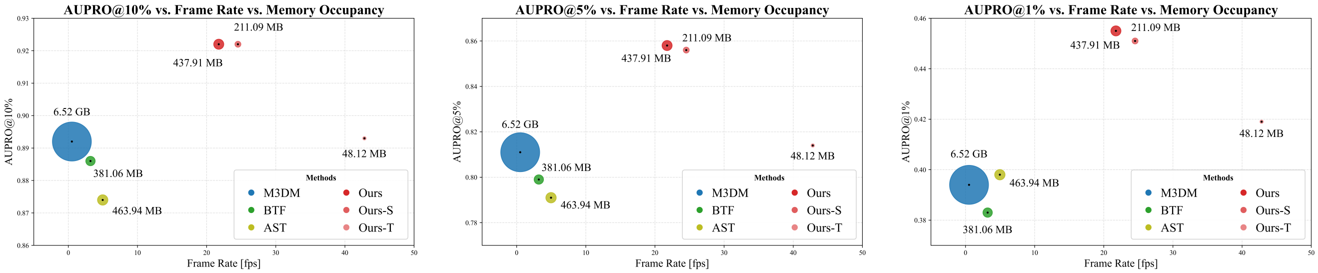

Figure 3. **Performance, speed and memory occupancy of Multimodal Anomaly Detection methods.** The chart reports anomaly segmentation performance on MVTec 3D-AD according to different AUPRO variants (from left to right: AUPRO@10%, AUPRO@5%, AUPRO@1%) vs. inference speed (Frame Rate on an NVIDIA 4090 GPU). The size of the symbols is proportional to memory occupancy at inference time.

0.1, 0.05, and 0.01, referred to as AUPRO@10%, AUPRO@5% and AUPRO@1%, respectively. As illustrated in Fig. 3, our proposal consistently provides better performance (*i.e*., higher AUPRO) than previous Multimodal AD methods across all the considered variants of the AUPRO metric while running much faster and requiring way less memory. In particular, the performance gap is higher for the more challenging variants of the AUPRO.

B. Feature Mapping Networks

We investigate the use of alternative network architectures to implement the Feature Mapping functions, namely: (i) MLP Encoder-Decoder, (ii) MLP Projection, *i.e*. the architecture described in the main paper, and (iii) Convolutional Encoder-Decoder.

The MLP Encoder-Decoder architecture comprises an encoding stage and a decoding stage, each consisting of two layers, along with an extra bottleneck layer between these two stages. The input layer in the encoding stage has a number of neurons equal to the dimensionality of the input feature space, while the last layer in the decoding stage has a number of neurons equal to the dimensionality of the output feature space. Between each pair of successive layers, but for the bottleneck layer, the number of neurons is either halved (in the encoding stage) or doubled (in the decoding stage). Accordingly, in our setup, we have $[768, 384, 192, 192, 384, 1152]$ neurons in each layer for $\mathcal{M}_{2D\rightarrow 3D}$, and $[1152, 576, 288, 288, 576, 768]$ neurons in each layer for $\mathcal{M}_{3D\rightarrow 2D}$. In both networks, all but the last layer employ GeLU activations.

As to MLP Projection architecture, we refer to shallow MLPs consisting of three layers, with GeLU activations but in the last one. The input layer has a number of neurons equal to the dimensionality of the input feature space, while the last layer has a number of neurons equal to the dimensionality of the output feature space. The intermediate layer has a number of

Metric	*Bagel*	*Cable Gland*	*Carrot*	*Cookie*	*Dowel*	*Foam*	*Peach*	*Potato*	*Rope*	*Tire*	**Mean**
					MLP Encoder-Decoder						
I-AUROC	0.993	0.858	**0.992**	0.988	0.985	0.911	0.959	0.866	0.986	0.864	0.940
AUPRO@30%	**0.979**	0.959	**0.982**	0.940	0.946	0.960	0.980	0.982	0.972	**0.981**	0.968
AUPRO@10%	**0.938**	0.882	0.946	0.890	0.843	0.883	0.941	0.946	0.918	0.942	0.913
AUPRO@5%	0.879	0.791	0.893	0.830	0.749	0.797	0.883	0.892	0.853	0.884	0.845
AUPRO@1%	0.467	0.385	0.487	**0.455**	0.385	0.395	0.466	0.480	0.451	0.466	0.444
Frame Rate (fps)											**25.769**
Memory (MB)											**369.856**
					MLP Projection (main paper)						
I-AUROC	0.990	**0.894**	0.986	0.989	0.980	0.916	0.951	**0.916**	0.986	0.886	0.949
AUPRO@30%	**0.979**	0.963	**0.982**	0.940	0.944	0.961	0.980	**0.983**	0.972	0.980	0.968
AUPRO@10%	0.937	0.892	**0.947**	0.890	0.838	0.885	0.940	**0.948**	0.918	0.941	0.914
AUPRO@5%	0.878	0.806	**0.894**	0.830	0.742	0.799	0.882	**0.897**	0.853	0.882	0.846
AUPRO@1%	**0.469**	0.402	0.486	0.450	0.380	0.397	0.463	**0.490**	0.453	0.463	0.445
Frame Rate (fps)											21.755
Memory (MB)											437.911
					Convolutional Encoder-Decoder						
I-AUROC	**0.997**	0.866	0.990	**0.993**	**0.989**	**0.927**	**0.979**	0.897	**0.990**	**0.918**	**0.955**
AUPRO@30%	**0.979**	**0.965**	**0.982**	**0.941**	**0.948**	0.969	**0.982**	**0.983**	**0.977**	**0.981**	**0.971**
AUPRO@10%	**0.938**	**0.897**	**0.947**	**0.893**	**0.847**	**0.906**	**0.945**	**0.948**	**0.931**	**0.944**	**0.920**
AUPRO@5%	**0.880**	**0.813**	**0.894**	**0.834**	**0.756**	**0.820**	**0.891**	0.896	**0.872**	**0.889**	**0.855**
AUPRO@1%	**0.469**	**0.409**	**0.488**	0.453	**0.393**	**0.409**	**0.477**	0.488	**0.467**	**0.473**	**0.453**
Frame Rate (fps)											9.906
Memory (MB)											2780.690

Table 1. Results on MVTec 3D-AD, Models trained for 50 epochs. Best results in **bold**, runner-ups underlined.

neurons equal to the mean between the dimensionality of the input and output features. Thus, as also reported in the main paper, in our setup the three layers in $\mathcal{M}_{2D \rightarrow 3D}$ have 768, 960 and 1152 neurons each, while the three layers of $\mathcal{M}_{3D \rightarrow 2D}$ have 1152, 960 and 768 neurons each.

Finally, unlike the previous two architectures which ingest individual feature vectors, the Convolutional Encoder-Decoder receives input tensors of spatial size $H \times W$ (with D_{2D} and D_{3D} channels for $\mathcal{M}_{2D \rightarrow 3D}$ and $\mathcal{M}_{3D \rightarrow 2D}$, respectively). The architecture follows a UNet-like structure without skip-connections, with two 3x3 convolutional layers followed by 2x2 max-pooling in the encoder stage and one 3x3 conv followed by a 2x2 transpose convolution in the decoding stage. All layers except the last one employ ReLU activations. The number of channels is kept equal to the input one up to the last layer, where it is modified so as to match the dimensionality of output feature space (*i.e.* from D_{2D} and D_{3D} for $\mathcal{M}_{2D \rightarrow 3D}$ and from D_{3D} and D_{2D} for $\mathcal{M}_{3D \rightarrow 2D}$.

For this new set of experiments, we follow the same training protocol as defined in the main paper. The results on MVTec 3D-AD are reported in Tab. 1, and show that the Convolutional Encoder-Decoder architecture provides slightly superior performance. However, despite its enhanced performance, it operates at a significantly slower inference rate, namely 9.906 fps, in contrast to the 21.755 fps achieved by our base model which is based on the MLP Projection architecture. Furthermore, the Convolutional Architecture requires six times more memory compared to our base model, *e.g.*, 2780.690 MB compared to 437.911 MB. Thus, we are led to prefer the performance vs efficiency (both speed and memory) trade-off provided by the MLP Projection architecture.

C. Feature Extractors

The ever-increasing availability of frozen Transformer-based RGB feature extractors trained on large data corpora has motivated us to explore alternatives to DINO ViT-B/8, such as, in particular, the ViT-B/16 used in SAM [4], the ViT-B/16 used in CLIP [6], and the ViT-B/14 used in DINO-v2 [5]. Results obtained on MVTec 3D-AD with the different 2D Feature Extractors are reported in Tab. 3. Interestingly, DINO and DINO-v2 exhibit much better performance than other feature extractors, which hints at - and may foster further investigation on - the benefits of foundation models trained via self-supervised contrastive learning in industrial AD.

D. Additional Quantitative Results

In this section, we report the class-wise anomaly detection and segmentation results for some of the experiments discussed in the main paper, considering also the additional FPR thresholds to compute the AUPRO introduced in Sec. A.

Metric	*Bagel*	*Cable Gland*	*Carrot*	*Cookie*	*Dowel*	*Foam*	*Peach*	*Potato*	*Rope*	*Tire*	Mean
					Ψ_{2D}						
I-AUROC	0.937	0.864	0.984	0.951	**0.984**	0.789	0.915	0.736	0.968	0.825	0.895
AUPRO@30%	0.960	0.966	0.979	0.884	0.911	0.916	0.981	0.974	0.958	0.971	0.950
AUPRO@10%	0.896	0.906	0.937	0.813	0.741	0.783	0.942	0.922	0.878	0.913	0.873
AUPRO@5%	0.819	0.834	0.874	0.738	0.624	0.675	0.884	0.844	0.789	0.841	0.792
AUPRO@1%	0.410	0.427	0.456	0.371	0.311	0.326	0.468	0.410	0.401	0.429	0.401
					Ψ_{3D}						
I-AUROC	0.948	0.770	0.968	0.981	0.937	**0.893**	0.694	0.909	0.939	0.812	0.885
AUPRO@30%	0.967	0.922	0.981	0.926	0.919	0.965	0.965	0.981	0.963	0.976	0.956
AUPRO@10%	0.903	0.782	0.943	0.871	0.764	0.899	0.894	0.943	0.892	0.928	0.882
AUPRO@5%	0.817	0.664	0.887	0.806	0.661	0.812	0.793	0.887	0.818	0.858	0.800
AUPRO@1%	0.402	0.302	0.474	0.443	0.341	0.389	0.338	0.474	0.431	0.437	0.403
					$\Psi_{2D} + \Psi_{3D}$						
I-AUROC	0.980	**0.893**	**0.991**	**0.996**	0.980	0.844	**0.970**	0.876	0.966	0.894	0.939
AUPRO@30%	0.969	0.968	0.980	0.904	0.914	0.958	**0.982**	0.977	0.961	0.977	0.959
AUPRO@10%	0.917	0.912	0.941	0.853	0.749	0.877	**0.945**	0.932	0.886	0.931	0.894
AUPRO@5%	0.852	**0.844**	0.882	0.799	0.638	0.784	**0.890**	0.864	0.806	0.869	0.823
AUPRO@1%	0.448	**0.439**	0.468	0.462	0.323	0.384	**0.478**	0.439	0.424	0.456	0.432
					$\max(\Psi_{2D}, \Psi_{3D})$						
I-AUROC	0.937	0.865	0.984	0.951	0.983	0.789	0.915	0.736	0.968	0.825	0.895
AUPRO@30%	0.960	0.966	0.979	0.884	0.911	0.916	0.981	0.974	0.958	0.971	0.950
AUPRO@10%	0.896	0.906	0.937	0.813	0.741	0.783	0.942	0.922	0.878	0.913	0.873
AUPRO@5%	0.819	0.834	0.874	0.738	0.624	0.675	0.884	0.844	0.789	0.841	0.792
AUPRO@1%	0.410	0.428	0.456	0.371	0.311	0.326	0.468	0.410	0.401	0.429	0.401
					$\Psi_{2D} \cdot \Psi_{3D}$						
I-AUROC	**0.994**	0.888	0.984	0.993	0.980	0.888	0.941	**0.943**	**0.980**	**0.953**	**0.954**
AUPRO@30%	**0.979**	**0.972**	**0.982**	**0.945**	**0.950**	**0.968**	0.980	**0.982**	**0.975**	**0.981**	**0.971**
AUPRO@10%	**0.937**	**0.917**	**0.947**	**0.897**	**0.855**	**0.906**	0.942	**0.947**	**0.926**	**0.944**	**0.922**
AUPRO@5%	**0.877**	0.843	**0.894**	**0.840**	**0.765**	**0.828**	0.884	**0.894**	**0.865**	**0.889**	**0.858**
AUPRO@1%	**0.459**	0.431	**0.485**	**0.469**	**0.394**	**0.413**	0.468	**0.487**	**0.464**	**0.476**	**0.455**

Table 2. **Aggregation analysis.** Best results in **bold**, runner-ups underlined.

$\mathcal{F}_{2D}$	**I-AUROC**	**P-AUROC**	**AUPRO@30%**	**AUPRO@1%**
DINO [2]	0.949	**0.992**	**0.968**	**0.445**
SAM [4]	0.792	0.973	0.906	0.311
CLIP [6]	0.833	0.984	0.942	0.346
DINO-v2 [5]	**0.958**	**0.992**	0.964	0.437

Table 3. **2D Feature Extractor Alternatives.** Results on MVTec 3D-AD. Best results in **bold**. Networks are trained for 50 epochs.

Metric	*Bagel*	*Cable Gland*	*Carrot*	*Cookie*	*Dowel*	*Foam*	*Peach*	*Potato*	*Rope*	*Tire*	**Mean**
					Ours						
I-AUROC	**0.994**	**0.888**	**0.984**	**0.993**	0.980	0.888	0.941	0.943	0.980	**0.953**	0.954
AUPRO@30%	0.979	**0.972**	**0.982**	0.945	0.950	0.968	0.980	0.982	0.975	0.981	0.971
AUPRO@10%	0.937	**0.917**	**0.947**	0.897	0.855	0.906	0.942	0.947	0.926	0.944	0.922
AUPRO@5%	0.877	**0.843**	0.894	0.840	0.765	0.828	0.884	0.894	0.865	0.889	0.858
AUPRO@1%	0.459	**0.431**	0.485	0.469	0.394	0.413	0.468	0.487	0.464	0.476	0.455
					Ours-M						
I-AUROC	0.988	0.875	**0.984**	0.992	**0.997**	**0.924**	**0.964**	**0.949**	0.979	0.950	**0.960**
AUPRO@30%	**0.980**	0.966	**0.982**	0.947	0.959	0.967	**0.982**	**0.983**	**0.976**	**0.982**	**0.972**
AUPRO@10%	**0.941**	0.901	**0.947**	0.899	0.880	0.901	**0.945**	**0.949**	**0.930**	**0.947**	**0.924**
AUPRO@5%	**0.884**	0.817	**0.895**	0.842	0.798	0.823	**0.890**	**0.898**	**0.872**	**0.893**	**0.861**
AUPRO@1%	**0.480**	0.398	0.490	0.467	**0.413**	0.408	**0.481**	**0.494**	**0.468**	**0.488**	**0.459**
					Ours-S						
I-AUROC	0.983	0.878	0.973	0.992	0.987	0.913	0.900	0.936	0.981	0.941	0.948
AUPRO@30%	0.978	0.960	**0.982**	**0.948**	**0.960**	**0.972**	0.977	**0.983**	**0.976**	0.981	**0.972**
AUPRO@10%	0.936	0.882	**0.947**	0.900	**0.884**	**0.918**	0.932	**0.949**	0.929	0.943	0.922
AUPRO@5%	0.874	0.782	0.894	0.843	**0.800**	**0.845**	0.864	**0.898**	0.870	0.886	0.856
AUPRO@1%	0.461	0.379	**0.492**	0.479	0.411	**0.429**	0.430	**0.494**	0.467	0.472	0.451
					Ours-T						
I-AUROC	0.948	0.784	0.946	0.985	0.946	0.855	0.815	0.932	**0.989**	0.794	0.899
AUPRO@30%	0.977	0.903	0.981	0.950	0.945	0.956	0.973	**0.983**	0.973	0.973	0.961
AUPRO@10%	0.932	0.736	0.944	**0.901**	0.838	0.873	0.919	**0.949**	0.920	0.918	0.893
AUPRO@5%	0.867	0.612	0.889	**0.844**	0.729	0.773	0.839	0.897	0.856	0.838	0.814
AUPRO@1%	0.449	0.267	0.487	**0.487**	0.364	0.369	0.395	0.491	0.462	0.421	0.419

Table 4. **Layers Pruning analysis.** Best results in **bold**, runner-ups underlined.

Method	*Bagel*	*Cable Gland*	*Carrot*	*Cookie*	*Dowel*	*Foam*	*Peach*	*Potato*	*Rope*	*Tire*	**Mean**
BTF [3]	0.996	0.992	0.997	0.994	0.981	0.974	0.996	0.998	0.994	0.995	0.992
AST [7]	-	-	-	-	-	-	-	-	-	-	0.976
M3DM [8]	0.995	0.993	0.997	0.985	0.985	0.984	0.996	0.994	0.997	0.996	0.992
Ours	0.997	0.992	0.999	0.972	0.987	0.993	0.998	0.999	0.998	0.998	0.993

Table 5. P-AUROC on MVTec 3D-AD dataset in comparison with state-of-the-art models.

In particular, Tab. 2 provides a detailed view of the results for the *Aggregation* function introduced in Sec. 3.3 of the main paper. As already highlighted in the evaluation summarized in Tab. 6 and discussed in Sec. 5 of the main paper, the product aggregation achieves the best results across most of the classes except for one class, *i.e.*, *Peach*, which shows higher results using the sum aggregation. These results further support our choice of relying on the product function, which realizes a logical AND between the discrepancies found in the individual modalities, as preferred aggregation approach.

In addition, Tab. 4 reports the detailed results for the *Layers Pruning* technique. As described in Sec. 3.4 of the main paper, to obtain lighter versions of our framework, we prune both feature extractors after the 1st, 4th, and 8th layer to obtain *Tiny*, *Small*, and *Medium* architectures, referred to as Ours-T, Ours-S and Ours-M. Thus, Tab. 4 extends the evaluation summarized in Tab. 5 and discussed in Sec. 5 of the main paper. It is worth noticing how Ours-M achieves the best results in both detection and segmentation. We also highlight that Ours obtains the second-best results in all average metrics.

For the sake of completeness, we also report in Tab. 5 the P-AUROC results on the MVTec 3D-AD dataset. As already anticipated in Sec. 5 of the main paper, this metric is mostly saturated since every method reaches the same very high results for each class.

As regards the Eyecandies dataset, we provide a detailed view of the results for each class in Tab. 6, also considering different FPR thresholds. It is worth highlighting that the original results provided by M3DM [8] were obtained by training on a subset of the train set of Eyecandies, mostly due to the limitations caused by the memory bank resource requirements. To achieve more comparable results, we retrained M3DM [8] on the full training set and reevaluated the benchmark, denoted as M3DM* in Tab. 6.

Generally, we note that features from deeper layers deliver higher contextualizations, thus enabling our cross-modal mapping to perform anomaly detection better, for the reasons highlighted in Sec. 3 of the main paper. However, some literature findings suggest that, in self-supervised learning, features from slightly shallower layers may turn out more task agnostic, i.e. exhibit a better ability to generalize to a wider range of downstream tasks. Thus, we argue that the above considerations may explain the slightly different performance between Ours and Ours-M in the considered datasets. Overall, we suggest the simplest and most general approach of keeping the whole Transformer-based feature extractors (i.e. Ours) as the default choice in our framework.

	Method	*Can. C.*	*Cho. C.*	*Cho. P.*	*Conf.*	*Gum. B.*	*Haz. T.*	*Lic. S.*	*Lollip.*	*Marsh.*	*Pep. C.*	**Mean**
I-AUROC	RGB-D [1]	0.529	0.861	0.739	0.752	0.594	0.498	0.679	0.651	0.838	0.750	0.689
	RGB-cD-n [1]	0.596	0.843	0.819	0.846	0.833	0.550	0.750	0.846	0.940	0.848	0.787
	M3DM [8]	0.624	**0.958**	**0.958**	**1.000**	**0.886**	0.758	**0.949**	0.836	**1.000**	**1.000**	**0.897**
	M3DM* [8]	0.597	0.954	0.931	0.990	0.883	0.666	0.923	0.888	0.995	**1.000**	0.882
	AST [7]	0.574	0.747	0.747	0.889	0.596	0.617	0.816	0.841	0.987	0.987	0.780
	Ours	**0.680**	0.931	0.952	0.880	0.865	**0.782**	0.917	0.840	0.998	0.962	0.881
	Ours-M	0.645	0.936	0.914	0.901	0.845	0.747	0.877	**0.904**	0.992	0.885	0.865
P-AUROC	RGB-D [1]	0.973	0.927	0.958	0.945	0.929	0.806	0.827	0.977	0.931	0.928	0.920
	RGB-cD-n [1]	0.980	0.979	**0.982**	0.978	0.951	0.853	0.971	0.978	0.985	0.967	0.962
	M3DM [8]	0.974	**0.987**	0.962	**0.998**	0.966	0.941	0.973	0.984	**0.996**	0.985	**0.977**
	M3DM* [8]	0.968	0.986	0.964	**0.998**	**0.976**	0.928	**0.976**	**0.988**	**0.996**	**0.995**	**0.977**
	AST [7]	0.763	0.960	0.911	0.969	0.788	0.837	0.918	0.924	0.983	0.968	0.902
	Ours	0.983	0.982	0.964	0.989	0.949	**0.946**	0.969	0.980	0.995	0.987	0.974
	Ours-M	**0.985**	0.984	0.961	0.986	0.958	0.937	0.968	0.981	0.994	0.978	0.973
AUPRO@30%	M3DM [8]	0.906	**0.923**	0.803	**0.983**	0.855	0.688	**0.880**	0.906	**0.966**	0.955	0.882
	M3DM* [8]	0.889	0.921	0.808	0.982	**0.889**	0.675	0.872	0.901	0.964	**0.973**	**0.887**
	AST [7]	0.514	0.835	0.714	0.905	0.587	0.590	0.736	0.769	0.918	0.878	0.744
	Ours	0.942	0.902	**0.831**	0.965	0.875	0.762	0.791	**0.913**	0.939	0.949	**0.887**
	Ours-M	**0.943**	0.892	0.795	0.962	0.871	**0.779**	0.767	0.909	0.944	0.935	0.880
AUPRO@10%	M3DM* [8]	0.677	**0.836**	0.698	**0.947**	**0.754**	0.410	**0.732**	0.712	**0.913**	**0.924**	0.760
	AST [7]	0.285	0.709	0.545	0.770	0.404	0.350	0.584	0.544	0.770	0.744	0.570
	Ours	0.827	0.815	**0.731**	0.896	0.741	**0.550**	0.663	**0.739**	0.893	0.868	**0.772**
	Ours-M	**0.829**	0.814	0.683	0.886	0.742	0.564	0.666	0.728	0.898	0.830	0.764
AUPRO@5%	M3DM* [8]	0.479	**0.759**	0.626	**0.894**	0.655	0.300	**0.634**	**0.562**	**0.849**	**0.861**	0.661
	AST [7]	0.173	0.592	0.421	0.635	0.288	0.242	0.461	0.378	0.634	0.617	0.444
	Ours	**0.662**	0.750	**0.653**	0.801	**0.657**	0.427	0.609	0.552	0.838	0.796	**0.675**
	Ours-M	0.661	0.747	0.611	0.792	0.665	**0.446**	0.619	0.518	0.840	0.751	0.665
AUPRO@1%	M3DM* [8]	0.166	0.388	0.329	**0.486**	0.315	0.131	0.323	**0.258**	**0.462**	**0.454**	0.331
	AST [7]	0.035	0.230	0.129	0.234	0.092	0.069	0.139	0.090	0.255	0.224	0.149
	Ours	**0.229**	**0.397**	**0.345**	0.389	**0.353**	0.188	0.333	0.236	0.455	0.428	**0.335**
	Ours-M	0.223	0.389	0.333	0.395	0.348	**0.206**	**0.342**	0.225	0.452	0.385	0.330

Table 6. **Various metrics on the Eyecandies dataset for several multimodal AD methods.** Best results in **bold**, runner-ups underlined.

E. Additional Qualitative Results

In Fig. 4, we highlight some failure cases of this approach. For instance, in the first left row, we note that our method cannot detect the missing left part of the cookie. Nevertheless, we predict higher anomaly scores for the area adjacent to the defect. In the second left row, the potato presents a tiny defect on its body, while the anomaly map — although covering the defect correctly — predicts a much broader anomaly. In the first and second right rows, the candy cane and the hazelnut truffle present high-frequency 2D or 3D patterns that produce higher anomaly scores compared to the real defects.

Finally, in Fig. 5 and Fig. 6 we show some additional qualitative results for all the classes of the MVTec 3D-AD and Eyecandies datasets, respectively. It is possible to notice how M3DM [8] tends to present anomalies on a broader area, highlighting the outline of the underlying object, while our method presents a more localized and less disturbed anomaly map.

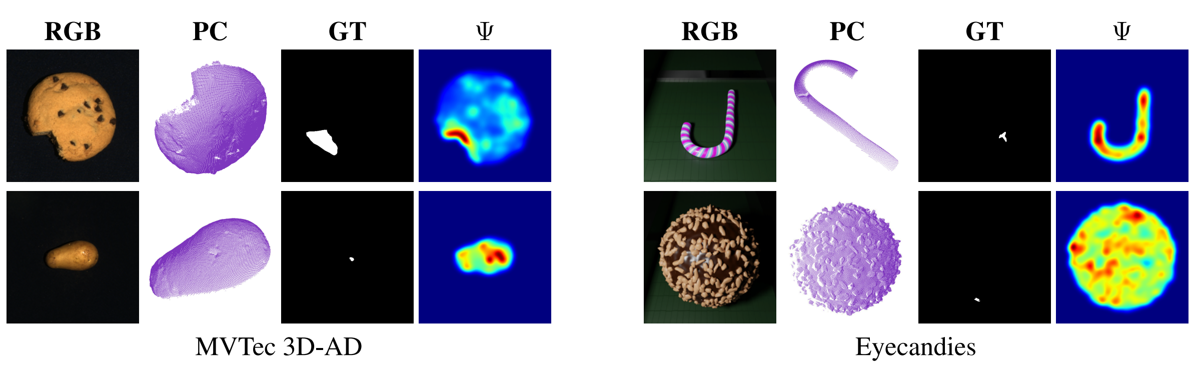

Figure 4. **Failure cases. Results on MVTec 3D-AD (left) and Eyecandies (right).**

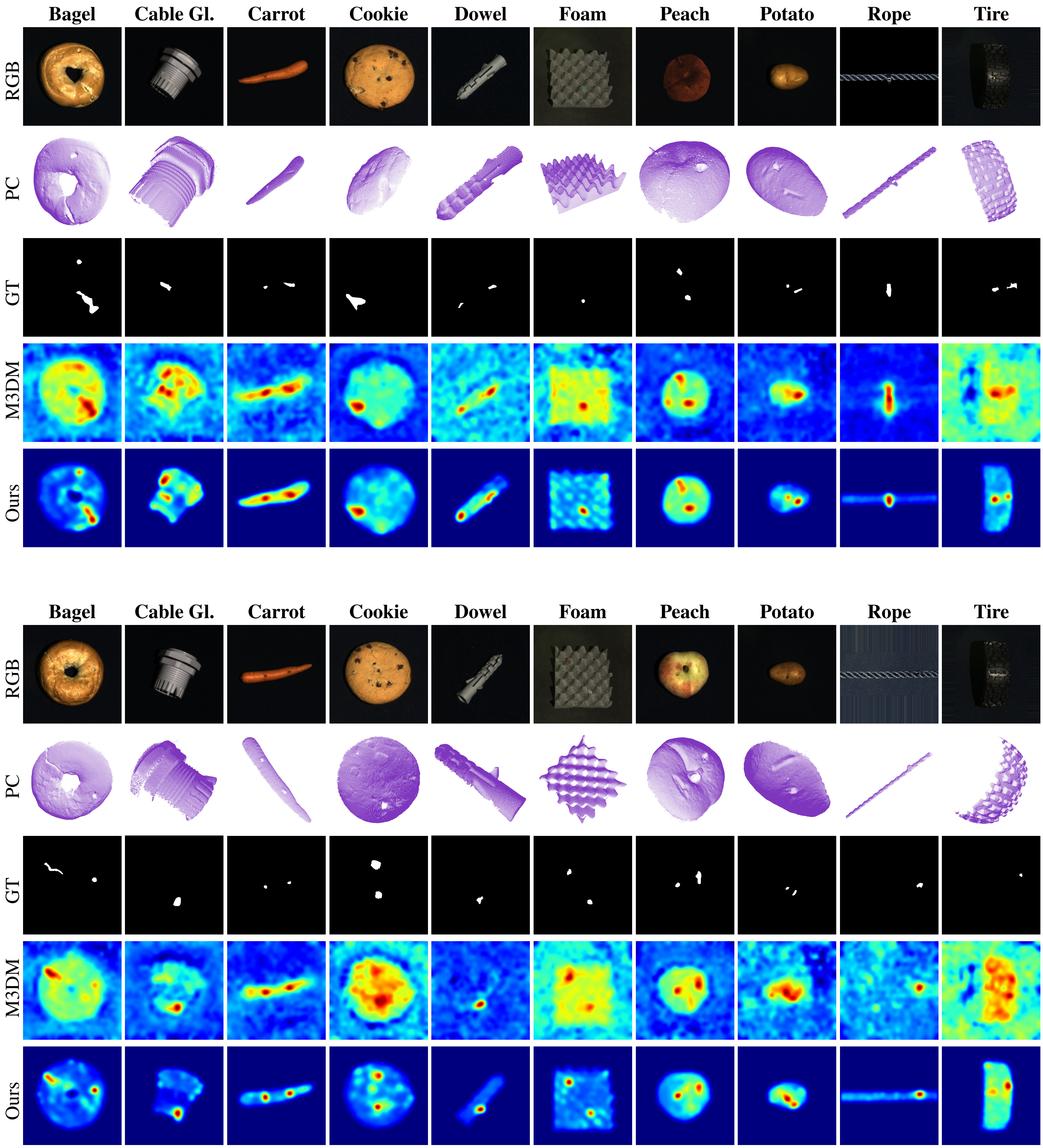

Figure 5. Qualitative results for each class of the MVTec 3D-AD dataset

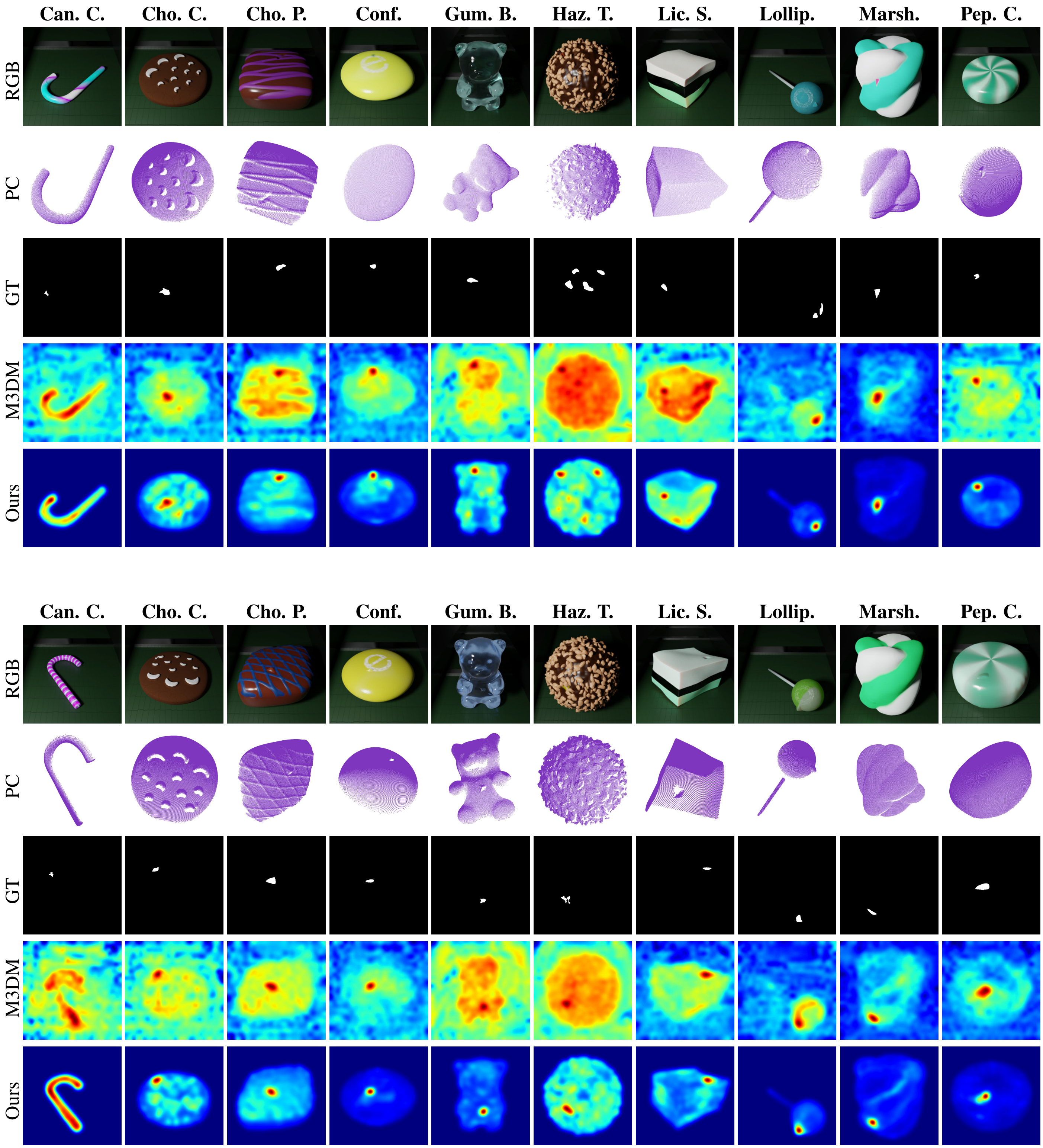

Figure 6. Qualitative results for each class of the Eyecandies dataset

References

[1] Luca Bonfiglioli, Marco Toschi, Davide Silvestri, Nicola Fioraio, and Daniele De Gregorio. The eyecandies dataset for unsupervised multimodal anomaly detection and localization. In *Proceedings of the 16th Asian Conference on Computer Vision (ACCV2022*, 2022. ACCV. 5

[2] 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. 4

[3] 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. 4

[4] 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. 3, 4

[5] 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. 3, 4

[6] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 3, 4

[7] Marco Rudolph, Tom Wehrbein, Bodo Rosenhahn, and Bastian Wandt. Asymmetric student-teacher networks for industrial anomaly detection. In *Winter Conference on Applications of Computer Vision (WACV)*, 2023. 4, 5

[8] 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. 4, 5