LogicAL: Towards logical anomaly synthesis for unsupervised anomaly localization

Supplementary Material

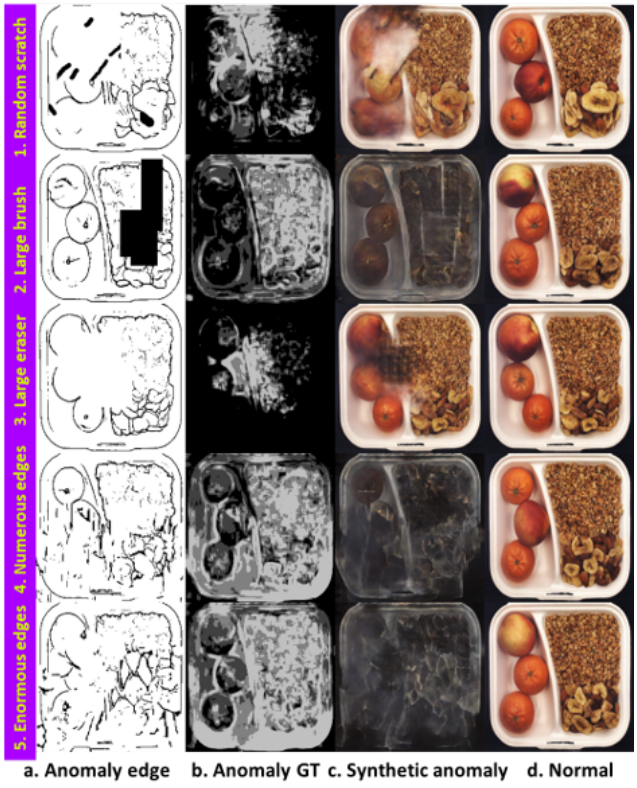

Figure 7. **Analysis of edge manipulation.** Improper anomaly edges cause collapse of edge-to-image generation.

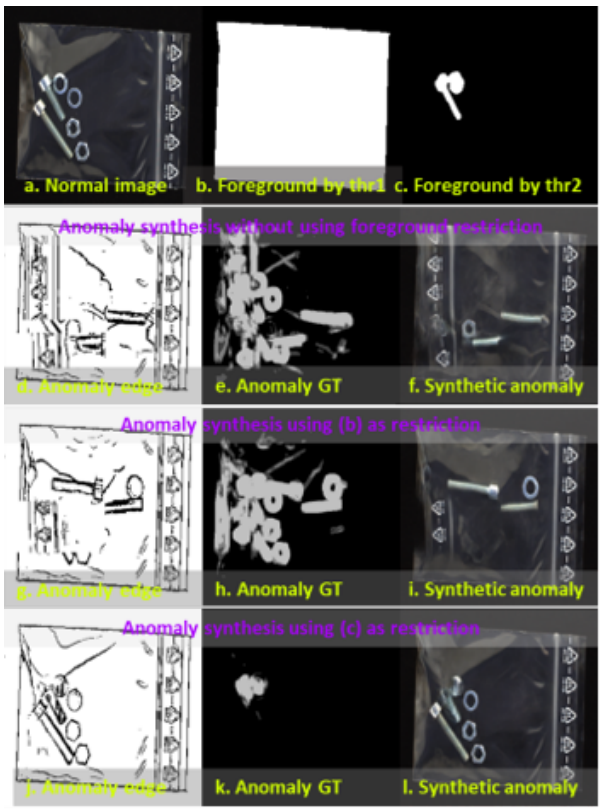

Figure 8. **Analysis of region selection.** Different foreground extractions may cause the inconsistency between original normal image and the synthetic anomaly image.

6. Ablation study

Edge manipulation. Given the pre-trained edge-to-image generator, it is supposed to generate anomaly images from the modified edge maps. Even though the generator is trained with carefully augmented data, it still produces global collapsed results if the input edge maps are out-of-distribution. Fig.7 illustrates the failure generations given anomaly edge maps from 5 different modifications. Overall, the modifications are based on the concerns of adding and/or removing different scales of edges. As shown in Fig.7(2), the edges forged by large brush are obviously different from the real edges. Fig.7(4)-(5) show that scale matters. Either numerous nor enormous modified edges make anomaly synthesis successfully. As the most of the original edges remained, the generator derives not perfect but better anomaly images from Fig.7(1) and (3). It indicates that a suitable portion of anomaly edges is the key to generate high quality anomaly images.

Region selection. Along with synthetic anomaly images, the ground truth for training anomaly localization module is simultaneously generated. Whether the ground truth conforms to logic is another issue that needs to be considered. Fig.8 demonstrates examples of this problem. Given a normal image Fig.8(a), two possible generated anomaly images are shown in Fig.8(f)and (i) respectively. Due to largely replacement of original edges, the generated anomaly images of Fig.6(f) and (i) are almost completely unrelated with the input normal image Fig.8(a). Considering the underline logic of normal data, the ground truth of Fig.8(f) should indicate the missing parts base on Fig.8(f) rather than the Fig.8(a). The anomaly localization module will be confused to learn from these kinds of training pairs. An extreme example is the replacement of entire edge map with another one.

Different with above mentioned situations, the synthetic anomaly image shown in Fig.8(l) having the consistency with the normal image and can provide a reasonable ground truth Fig.8(k) for the anomaly localization training. The

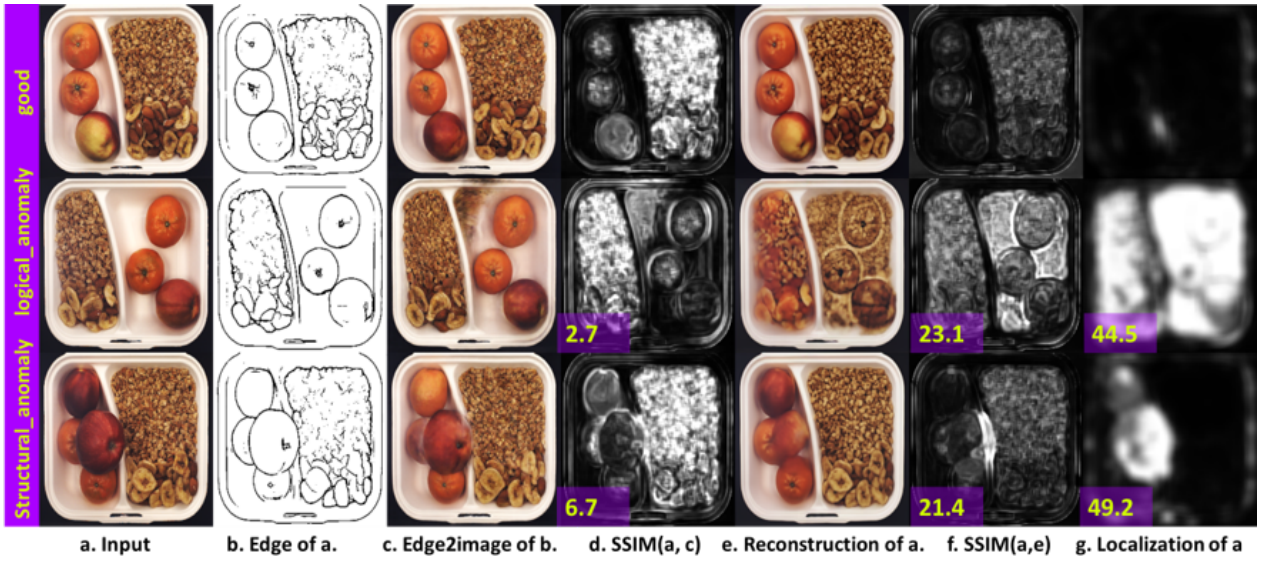

Figure 9. **Comparison of anomaly localization.** We compare our localization map (g) with the difference maps (d)(f) that are produced by calculating SSIM between generated images (c)(e) and input (a). (c) is generated by the edge-to-image generator. (e) is generated by our reconstruction sub-network. The pixel AUC-sPro of the class(breakfastBox) are illustrated in lower left corners.

success of Fig.8(l) is due to its effective foreground restriction, as shown in Fig.8(c), extracted with a larger threshold(thr2>thr1). Due to edges caused by background reflection, such as plastic packaging, it is difficult to precisely extract the foreground from JND map with a simple and costless method. The foreground extraction is not necessary to be perfect but have to remain parts of the key components to maintain consistency. Motivated by these observations, we construct anomaly edges based on region selection strategies that avoids generation collapse and maintains consistency.

Anomaly localization. Fig.9 illustrates the advance performance of our localization sub-network comparing directly using SSIM metric to spot anomaly regions. Fig.9c and Fig.9d indicate that the edge-to-image generator converts edge maps (b) to color images (c) with little sense of anomalies. It is barely impossible to use it indicate anomaly regions, especially the logical anomaly. On the contrary, our reconstruction images (e) reveal more vivid corruption in the anomaly regions.9e.

7. Loss functions

The total loss L consists of reconstruction L_{rec} and segmentation L_{seg} losses. For both JND map and normal image reconstruction, we not only use MSE loss to supervise the pixel-to-pixel recovering but also the structural similarity (SSIM) [24] loss to yield plausible local consistency.

$$L = L_{rec} + L_{seg} \tag{2}$$

$$L_{rec} = L_{img} + L_{jnd} + L_{edge} \tag{3}$$

$$L_{img} = L_2(I, I_{rec}) + L_s sim(I, I_{rec}) \tag{4}$$

$$L_{jnd} = L_2(J, J_{rec}) + L_s sim(J, J_{rec}) \tag{5}$$

where I and J are input normal image and corresponding JND map, I_{rec} and J_{rec} are output reconstructions.

Following PiDiNet [21], we adopt the annotator-robust loss function proposed in [16] for the reconstructed edge maps. For the ith pixel in the jth edge map with value p_i^j, the loss is calculated as:

$$L_{edge} = \begin{cases} \alpha \cdot (log(1 - p^j)), if\ y = 0 \\ 0, if\ 0 < y < \eta \\ \beta \cdot (logp^j), otherwise \end{cases} \tag{6}$$

where y is the ground truth edge probability generated by PiDiNet [21], η is the binary threshold, $\beta = 1.1$ is the percentage of negative pixel samples and $\alpha = 1.0$ is the percentage of positive pixel samples.

To handle the unbalance of normal and anomaly, we use focal loss [14] to supervise the predicted anomaly localization.

$$L_{seg} = L_{focalloss}(M, M_{seg}) \tag{7}$$

8. Anomaly synthesis

Fig.10-Fig.12 visualize the synthetic anomalies of MVTecAD [1], VisA [34] and MADsim [33] datasets respectively. We randomly apply region selection and edge modification strategies to generate anomaly edge maps for all datasets. We also randomly apply colorjitter to VisA [34] and TPS warping to MADsim [33] datasets.

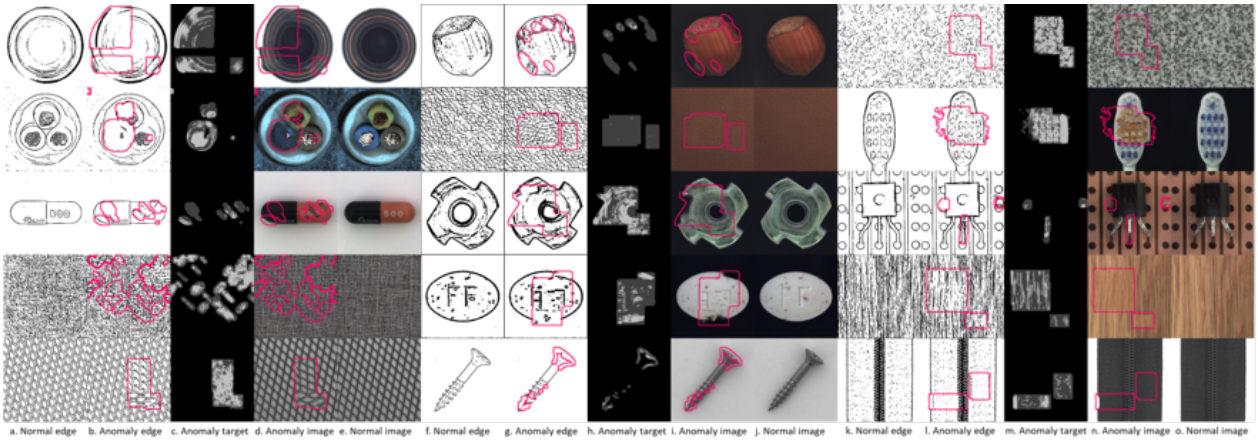

Figure 10. **Visualization of synthetic anomaly for MVTecAD [1].**

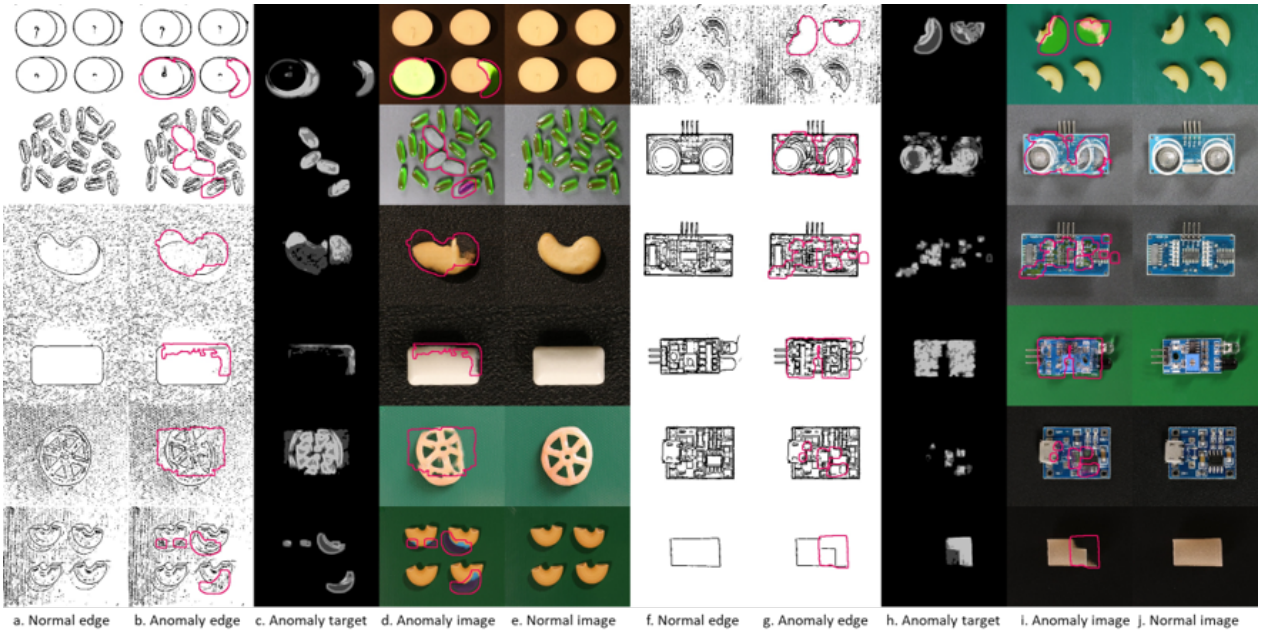

Figure 11. **Visualization of synthetic anomaly for VisA [34].**

9. Qualitative evaluation

Fig.13-Fig.14 visualize the synthetic anomalies of VisA [34] and MADsim [33] datasets respectively.

Figure 12. **Visualization of synthetic anomaly for MADsim [33].**

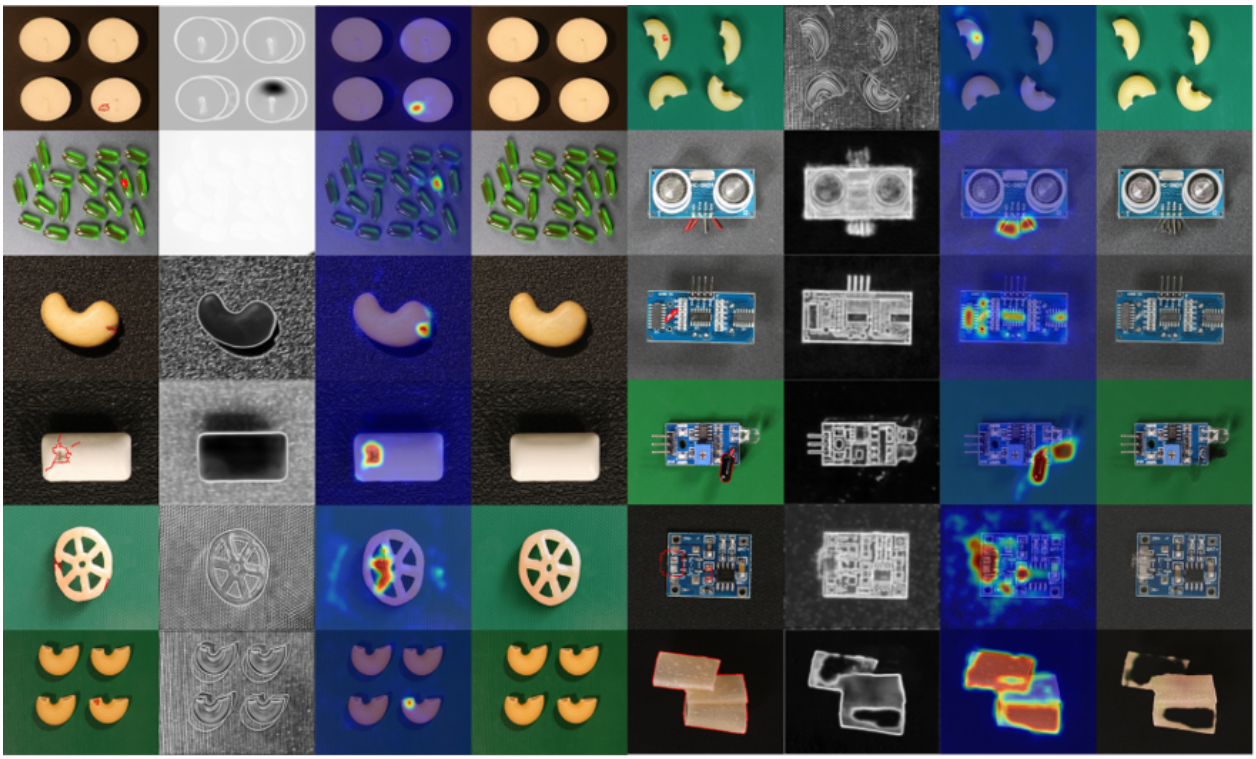

Figure 13. **Qualitative illustration of our anomaly detection results on VisA [34].**

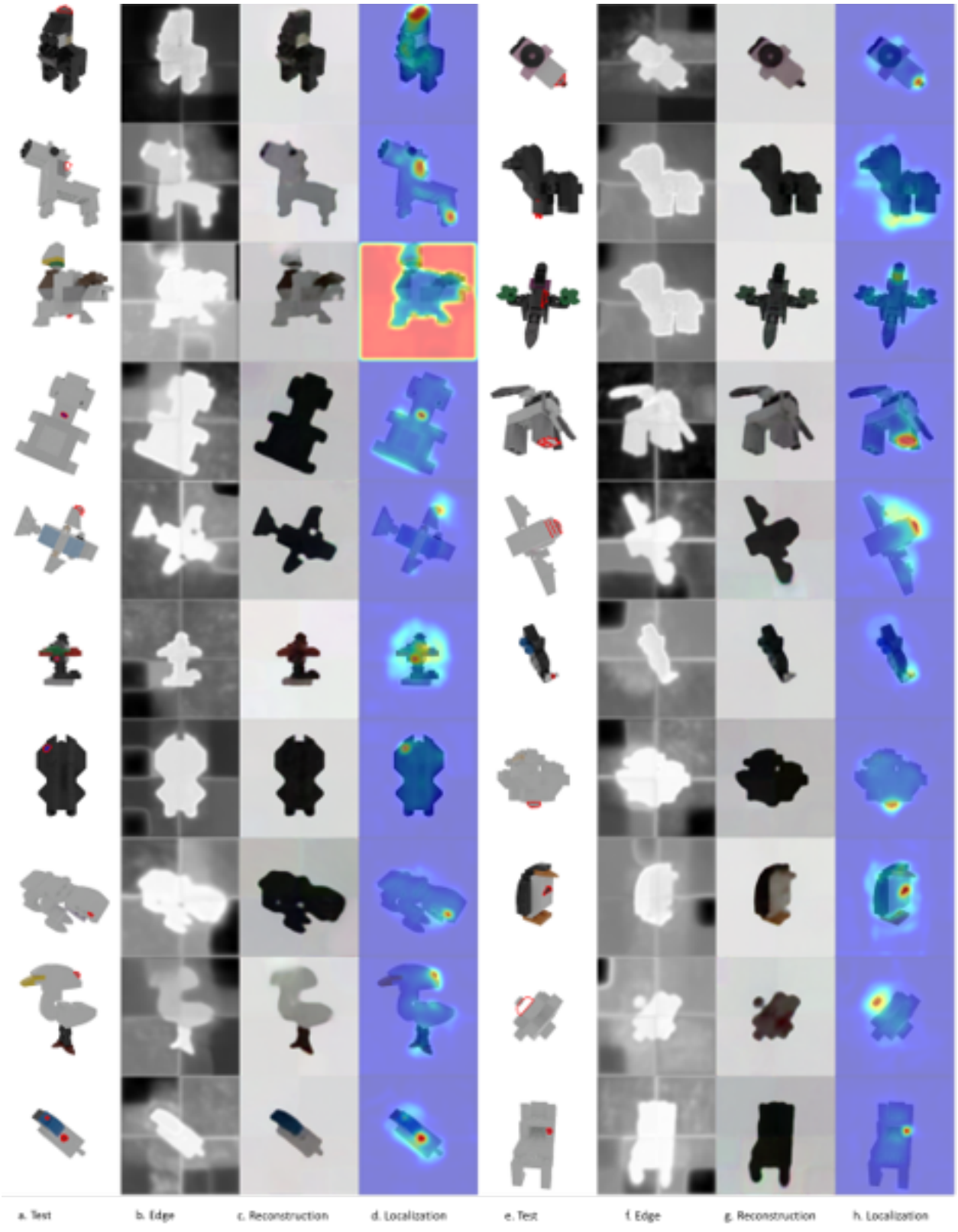

Figure 14. **Qualitative illustration of our anomaly detection results on MADsim [33].**