Unsupervised Surface Anomaly Detection with Diffusion Probabilistic Model Supplementary File

Abstract

In this supplementary file, we first present more details about our experiments, including more implementation details and results of the ablation experiments. For efficiency issues, we also provide a comparison based on different sampling steps. Finally, we provide quantitative comparisons with more methods for the anomaly detection task.

Additionally, more qualitative results of our DiffAD are shown to visualize the effectiveness of our method.

1. Implementation Details

In this section, we provide more details of our ablation experiments, including two diffusion-based *DiffAD* variants with different ways of adding conditions (*e.g.* *DiffAD*$_{f\&r}$ and *DiffAD*$_{c}$), and two variants with different inputs to the discriminative sub-network (*e.g.* *DiffAD*$_{no\_inter}$ and *DiffAD*$_{\tilde{x}_a\_inter}$). We provide more implementation details and the specific experimental results of each class.

1.1. Variants of conditions

Besides our noisy condition embedding, we also provide two alternative methods: (i) injecting noise to inputs and then reconstructing based on diffusion models (*DiffAD*$_{f\&r}$), (ii) concatenating the latent vector $\mathbf{c}$ of the simulated anomalous sample as the condition during training of diffusion models (*DiffAD*$_{c}$).

1.1.1 *DiffAD*$_{f\&r}$

DiffAD$_{f\&r}$ trains a diffusion model solely on normal samples without conditions. During the sampling stage, the test instance is first injected noise by a forward process. Then the noisy version of the test sample is taken as input and gradually denoised by the diffusion model. The anomalous features are damaged by the noise while some global information is retained, so the anomalous samples can be reconstructed to normal ones. However, the time step t of the forward process needs to be carefully selected, for a small t may retain anomalous features while a large t may lose too much information. Setting the total length T of the Markov chain to 1000, we choose $t = 800$ as the time step to generate noisy samples. We provide specific experiment results in Table 1 and 2.

Class		$D_{f\&r}$	D_c	D_{no}	$D_{\tilde{x}_a}$	*DiffAD*
texture	Carpet	97.9	94.5	96.5	98.2	98.3
	Grid	100	100	100	100	100
	Leather	100	100	100	99.7	100
	Tile	90.5	100	100	100	100
	Wood	99.8	100	99.7	99.8	100
object	Bottle	97.8	97.4	99.1	97.9	100
	Cable	84.1	78.1	88.5	90.0	94.6
	Capsule	91.0	94.2	92.4	95.2	97.5
	Hazelnut	99.6	100	92.5	98.4	100
	Metal Nut	99.9	98.1	97.4	99.4	99.5
	Pill	90.2	88.6	94.7	96.8	97.7
	Screw	89.3	93.7	94.8	96.6	97.2
	Toothbrush	99.4	99.7	98.3	99.7	100
	Transistor	81.3	93.0	92.6	89.8	96.1
	Zipper	98.2	99.9	99.9	99.8	100
Average		94.6	95.8	96.4	97.4	98.7

Table 1. Results for anomaly detection with AUROC metric on MVTec-AD, compared with other *DiffAD* variants.

1.1.2 *DiffAD*$_{c}$

Directly concatenating the input images as conditions during the training process of diffusion models, *DiffAD*$_{c}$ is capable of reconstructing most of the anomalous cases. However, for some hard cases especially ones with severe structural changes, some anomalous features cannot be modified well, leading to unsatisfying reconstruction performance, as shown in Figure 1.

1.2. Variants of inputs

Table 1 and 2 report the specific results of variants trained (i) without the interpolated channels (*DiffAD*$_{no\_inter}$), (ii) with decoding the latent vector $\mathbf{c}$ of the anomalous input into $\tilde{x}_a$ as the additional channels, *i.e.* $\lambda = 1$ (*DiffAD*$_{\tilde{x}_a\_inter}$) and (iii) with interpolated channels with $\lambda = 0.5$ (*DiffAD*). From the experiment results,

	Class	*DiffAD*$_{f\&r}$	*DiffAD*$_c$	*DiffAD*$_{no\_inter}$	*DiffAD*$_{\tilde{x}_a\_inter}$	*DiffAD*
texture	Carpet	95.5 / 64.6	94.6 / 54.2	96.6 / 55.2	96.4 / 66.2	98.1 / 74.1
	Grid	98.5 / 71.2	99.4 / 63.4	99.6 / 69.1	99.6 / 70.6	99.7 / 73.7
	Leather	97.4 / 71.2	97.8 / 61.1	98.6 / 69.5	97.6 / 58.0	99.1 / 73.7
	Tile	87.7 / 56.1	99.0 / 95.9	99.0 / 95.0	99.4 / 95.9	99.4 / 95.1
	Wood	95.3 / 78.2	95.7 / 68.1	94.2 / 66.2	97.1 / 71.3	96.7 / 80.0
object	Bottle	94.9 / 66.8	97.2 / 79.5	98.4 / 83.5	98.5 / 84.6	98.8 / 87.4
	Cable	87.5 / 36.8	81.6 / 27.9	92.7 / 39.5	96.0 / 59.9	96.8 / 64.9
	Capsule	89.3 / 35.5	79.6 / 28.5	93.6 / 47.4	93.5 / 43.9	98.2 / 54.4
	Hazelnut	97.1 / 69.7	99.1 / 77.4	96.8 / 61.7	99.1 / 76.6	99.4 / 85.9
	Metal Nut	97.6 / 89.4	98.3 / 90.0	98.5 / 91.9	99.2 / 94.8	99.1 / 94.4
	Pill	95.9 / 41.8	95.7 / 47.1	96.6 / 41.2	97.6/ 52.9	97.7 / 68.9
	Screw	87.2 / 35.2	95.4 / 54.6	98.4 / 54.6	98.7 / 50.3	99.0 / 58.5
	Toothbrush	96.6/ 46.9	98.6 / 62.1	98.8 / 68.0	97.8 / 60.6	99.2 / 70.1
	Transistor	85.8 / 32.8	78.5 / 31.0	93.7 / 56.6	89.8 / 40.6	93.7 / 60.2
	Zipper	97.6 / 63.4	98.5 / 75.6	98.8 / 77.3	98.8 / 78.0	99.0 / 77.8
Average		93.6 / 57.3	93.9 / 61.1	97.0 / 65.1	97.3 / 67.0	98.3 / 74.6

Table 2. Results for anomaly localization with AUROC / AP metric on MVTec-AD, compared with other *DiffAD* variants.

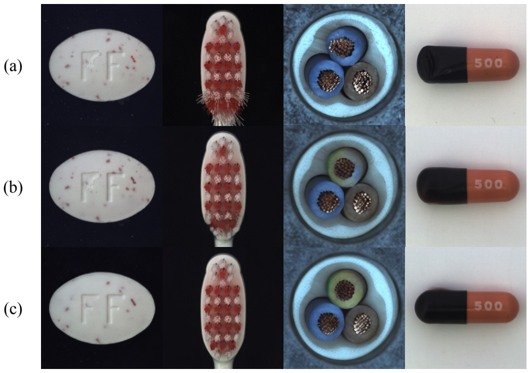

Figure 1. Visual comparisons between (a) the anomalous inputs; the reconstruction outputs of (b) *DiffAD*$_c$, and (c) *DiffAD*.

the interpolated channels play more critical roles in classes with high diversity, such as hazelnut and cable.

Class	[3]	[2]	[5]	[4]	[6]	Ours
Carpet	98.9	99.8	98.7	**100**	96.9	98.3
Grid	**100**	96.7	98.2	97.6	**100**	**100**
Leather	**100**	**100**	**100**	97.7	99.6	**100**
Tile	99.3	98.1	98.7	98.7	98.6	**100**
Wood	99.2	99.2	99.2	99.6	98.8	**100**
Bottle	**100**	99.9	**100**	**100**	**100**	**100**
Cable	95.0	92.7	99.5	**100**	96.8	94.6
Capsule	96.3	91.3	98.1	**99.3**	96.1	97.5
Hazelnut	99.9	92.0	**100**	96.8	99.9	**100**
Metal Nut	**100**	98.7	**100**	91.9	97.2	99.5
Pill	96.6	93.3	96.7	**99.9**	95.3	97.7
Screw	97.0	85.8	98.1	**99.7**	99.6	97.2
Toothbrush	99.5	96.1	**100**	95.2	99.8	**100**
Transistor	96.7	97.4	**100**	99.1	95.4	96.1
Zipper	98.5	90.3	98.8	98.5	99.8	**100**
Average	98.5	95.5	**99.1**	98.3	98.2	98.7

Table 3. Results for anomaly detection with AUROC metric on MVTec-AD, compared with more baseline methods.

2. Computational Complexity

2.1. Accuracy and training efficiency

The initial DDPM samples images by denoising noise step by step, leading to low inference speed. DDIM accelerates the sampling process, increasing the sampling speed by 10 to 50 times. The latent diffusion models shift to the latent space, further reducing training and sampling overheads. DDPM takes 1000 iterations to generate samples, while the latent diffusion model only takes 50 steps to achieve satisfactory results. In our *DiffAD*, the sampling steps of the reconstructive sub-network can be further reduced to raise efficiency. As shown in Table 4, even taking 10 steps to sample images in the training stage of the discriminative sub-network, our method can still achieve comparable results especially in some classes with relatively simple patterns such as some texture type classes. Therefore, the computation burden brought by diffusion models can be greatly reduced.

2.2. Inference Time

Inference time depends on the number of iterations during sampling, which is set at 5 during testing in our experiments. As shown in Table 5, our method strikes a good bal-

Steps		10		20		50	
Class		Det.	Loc.	Det.	Loc.	Det.	Loc.
texture	Carpet	98.1	97.9 / 72.3	98.0	97.9 / 73.5	98.3	98.1 / 74.1
	Grid	100	99.6 / 73.9	99.8	97.9 / 73.5	100	99.7 / 73.7
	Leather	100	98.7 / 66.8	100	98.6 / 66.4	100	99.1 / 73.7
	Tile	100	99.3 / 95.1	100	99.2 / 94.9	100	99.4 / 95.1
	Wood	100	96.3 / 78.7	100	96.1 / 77.9	100	96.7 / 80.0
object	Bottle	99.3	98.7 / 85.1	99.5	98.7 / 87.2	100	98.8 / 87.4
	Cable	93.6	95.8 / 61.1	92.5	96.0 / 58.4	94.6	96.8 / 64.9
	Capsule	96.9	95.7 / 47.5	98.3	96.9 / 48.9	97.5	98.2 / 54.4
	Hazelnut	99.8	99.4 / 85.6	100	99.3 / 84.9	100	99.4 / 85.9
	Metal Nut	98.2	99.2 / 94.2	98.4	99.1 / 94.4	99.5	99.1 / 94.4
	Pill	97.2	96.7 / 58.3	97.5	97.6 / 66.3	97.7	97.7 / 68.9
	Screw	96.8	98.0 / 57.3	96.8	98.3 / 59.3	97.2	99.0 / 58.5
	Toothbrush	99.4	98.9 / 67.5	99.4	98.9 / 68.3	100	99.2 / 70.1
	Transistor	93.6	90.9 / 53.2	94.1	92.6 / 57.5	96.1	93.7 / 60.2
	Zipper	99.7	99.0 / 79.0	100	99.0 / 77.5	100	99.0 / 77.8
Average		98.2	97.6 / 71.7	98.3	97.9 / 72.5	98.7	98.3 / 74.6

Table 4. Comparison with different sampling steps of the reconstructive sub-network.

Method	Times(s)	Det.	Loc.
DiffAD$_5$	0.10	98.7	**98.3 / 74.6**
DRAEM [7]	**0.01**	98.0	97.3 / 68.4
PaDim [2]	0.19	95.3	97.4 / 55.0
PatchCore-10% [5]	0.22	**99.0**	98.1 / 63.1

Table 5. Results for average inference time, detection (Det.) and localization (Loc.) on MVTec-AD with NVIDIA Tesla V100.

ance between efficiency and performance. Though slower than AE-based SOTA (DRAEM), our efficiency is superior to other non-AE SOTAs, and meets practical application requirements.

3. Additional Results

3.1. Quantitative Results

We have compared our method with reconstruction-based methods in the anomaly detection task. We provide more qualitative comparisons with other methods in Table 3, including distillation-based method: RDistillation [3]; representation-based methods: PaDim [2] and PatchCore [5]; flow-framework-based method: CFlow [4]; and a method based on score-based generative model: Score-DD [6]. Our method achieves comparable results to previous best-performing methods with 98.7% of AUROC.

3.2. Qualitative Results

Among the MVTec-AD [1] dataset, some given ground truth anomaly masks are not so accurate, resulting in an underestimated score in localization. We provide more qualitative examples in Figure 2. For example, the ground truth covers the entire surface of the pill (the 5th column), yet only the yellow dots are anomalous. Our predicted mask successfully distinguish the anomalies, but the difference with the ground truth mask increases the performance error.

References

[1] 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*, pages 9592–9600, 2019.

[2] Thomas Defard, Aleksandr Setkov, Angelique Loesch, and Romaric Audigier. Padim: a patch distribution modeling framework for anomaly detection and localization. In *Pattern Recognition. ICPR International Workshops and Challenges, Proceedings, Part IV*, pages 475–489. Springer, 2021.

[3] 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.

[4] 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.

[5] Karsten Roth, Latha Pemula, Joaquin Zepeda, Bernhard Schölkopf, Thomas Brox, and Peter Gehler. Towards total recall in industrial anomaly detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14318–14328, 2022.

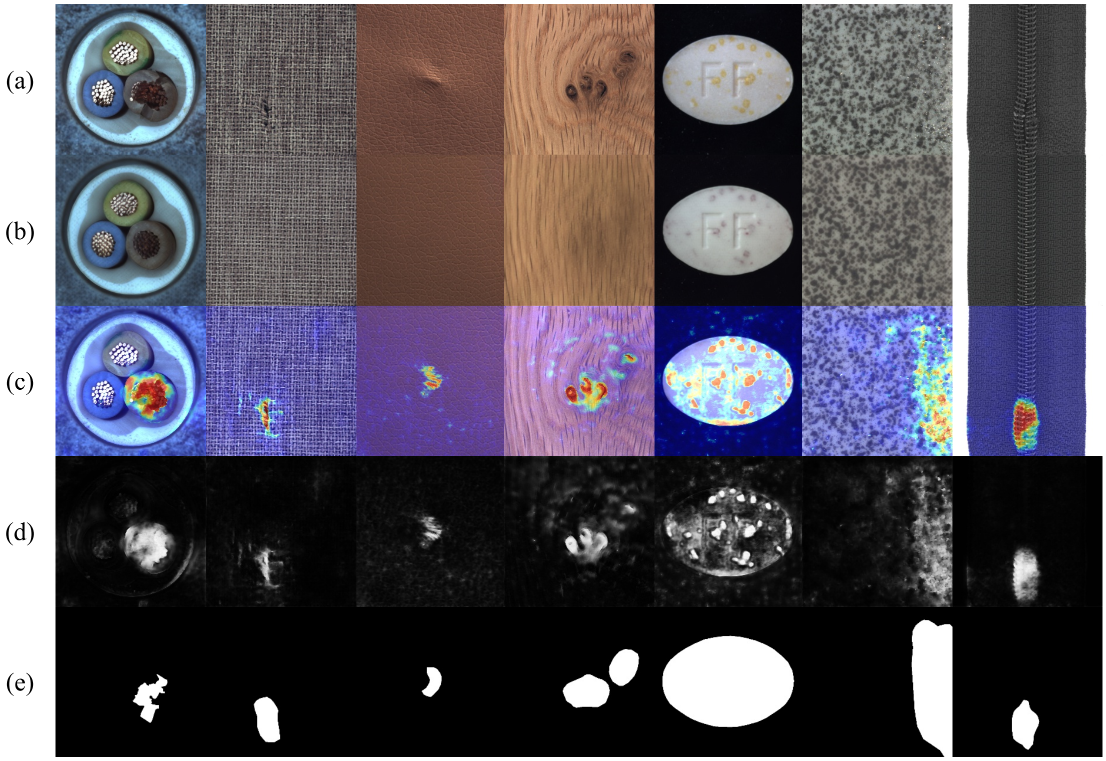

Figure 2. Qualitative examples. From top to bottom: the original anomalous input, our reconstruction, our predicted anomaly heat map, our predicted anomaly mask, and the ground truth mask. In these cases, the annotations are ambiguous and our predicted masks are more accurate than the given ground truth masks.

[6] Yapeng Teng, Haoyang Li, Fuzhen Cai, Ming Shao, and Siyu Xia. Unsupervised visual defect detection with score-based generative model. *arXiv preprint arXiv:2211.16092*, 2022.

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