Appendices

First, we give the formulation of unsupervised anomaly detection (UAD). Let $\mathcal{D}^{train} = \{(x_i, y_i)\}_{i=1}^{N}$, $y_i \in \{-\}$ be the training set of normal samples, where x_i is the i^{th} sample with its label y_i. During test, $\mathcal{D}^{test} = \{(x_i, y_i)\}_{i=1}^{M}$, $y_i \in \{-, +\}$ ($-$ for normal, $+$ for anomalous) contains both normal and anomalous images. Our goal is to train a model using $\mathcal{D}^{train}$ to identify $(x, +)$ from $(x, -)$ in $\mathcal{D}^{test}$, while localizing anomalous regions at the same time.

A Related Work

Epistemic methods are based on the assumption that the networks respond differently during inference between seen input and unseen input. Within this paradigm, *pixel reconstruction* methods assume that the networks trained on normal images can reconstruct anomaly-free regions well, but poorly for anomalous regions [1]. Auto-encoder (AE) [2; 1], variational auto-encoder (VAE) [3; 4], or generative adversarial network (GAN) [5; 6; 7] are used to restore normal pixels. However, *pixel reconstruction* models may also succeed in restoring unseen anomalous regions if they resemble normal regions in pixel values or the anomalies are barely noticeable [8; 9]. Therefore, *feature reconstruction* is proposed to construct features of pre-trained encoders instead of raw pixels [8; 10; 11]. To prevent the whole network from converging to a trivial solution, the parameters of the encoders are frozen during training [9]. In *feature distillation* [8; 12; 13], the student network is trained from scratch to mimic the output features of the pre-trained teacher network with the same input of normal images, also based on the similar hypothesis that the student trained on normal samples only succeed in mimicking features of normal regions.

Pseudo-anomaly methods generate handcrafted defects on normal images to imitate anomalies, converting UAD to supervised classification [14] or segmentation task [15]. Specifically, CutPaste [14] simulates anomalous regions by randomly pasting cropped patches of normal images. DRAEM [15] constructs abnormal regions using Perlin noise as the mask and another image as the additive anomaly. SimpleNet [16] introduces anomaly by injecting Gaussian noise in the pre-trained feature space. These methods deeply rely on how well the pseudo anomalies match the real anomalies [8], which makes it hard to generalize to different datasets.

Feature memory & modeling methods [17; 18; 19; 20; 21] memorize all (or their modeled distribution) normal features extracted by networks pre-trained on large-scale datasets and match them with test samples during inference. Since these methods require memorizing, processing, and matching nearly all features from training samples, they are computationally expensive in both training and inference, especially when the training set is large. There is a considerable semantic gap between large-scale natural images on which the frozen networks were pre-trained and the target UAD images in various modalities. Therefore, methods based on pre-trained networks struggle in transferring as the feature encoders are not optimized in the target domain.

There are a few studies addressing the adaptation problem in UAD settings. CFA [21] and SimpleNet[16] unanimously utilize a learnable linear layer to transform the output features of the frozen encoder. However, the linear layer may be insufficient for adaption and can hardly recover the domain-specific information in pre-trained features, especially when the target domain, e.g. CT and MRI, is remote from natural image domain. In OCC settings, a number of works focus on utilizing self-supervised pre-training approaches to learn a compact representation space for normal samples [22; 23; 24]. However, the performances are far from satisfactory in UAD tasks (I-AUROC under 90% on MVTec AD).

B Datasets

MVTec AD [25] is an industrial defect detection dataset, containing 15 sub-datasets (5 texture categories and 10 object categories) with a total of 3,629 normal images as the training set and 1,725 normal and anomalous images as the test set. Pixel-level annotations are available for anomalous images for evaluating anomaly segmentation. All images are resized to 256×256.

VisA [26] is an industrial defect detection dataset, containing 12 sub-datasets with a total of 9,621 normal and 1200 anomalous images. Pixel-level annotations are available for anomalous images for evaluating anomaly segmentation. We split it into training and test sets following the setting in [26]. All images are resized to 256×256.

OCT2017 is an optical coherence tomography (OCT) dataset [27]. The dataset is labeled into four classes: normal, drusen, DME (diabetic macular edema), and CNV (choroidal neovascularization). The latter three classes are considered anomalous. The training set contains 26,315 normal images and the test set contains 1,000 images. All images are resized to 256×256.

APTOS is a color fundus image dataset, available as the official training set of the 2019 APTOS blindness detection challenge [28]. The dataset contains 3,662 images with annotated grade 0-4 to indicate different severity of diabetic retinopathy, yielding 1,805 normal images (grade 0) and 1857 anomalous images (grade 1-4). We randomly selected 1,000 normal images as our training set and the rest 2,662 images as the test set. Images are preprocessed to crop the fundus region, and then resized to 256×256.

ISIC2018 is a skin disease dataset, available as task 3 of ISIC2018 challenge [29]. It contains seven classes. NV (nevus) is taken as the normal class and the rest of classes are taken as anomaly, following [30]. The training set contains 6705 normal images. The official validation set is used as our test set, which includes 193 images. Images are resized to 256×256 and then center cropped into 224×224 to remove redundant background. In addition, because the normal and anomalous images of ISIC2018 are different objects (lesions) instead of healthy objects and unhealthy objects, the task is more like one-class classification. Therefore, we take the mean (instead of maximum) value of $\mathcal{S}^{map}$ as the $\mathcal{S}^{img}$ in ReContrast and other reproduced methods.

C Complete Implementation Details

WideResNet50 [31] pre-trained on ImageNet [32] is utilized as the encoder by default. The decoder is the upside-down version of the encoder, exactly the same as RD4AD [8], i.e., the 3×3 convolution with stride 2 at the beginning of each layer is replaced by a 2×2 transposed convolution with stride 2. The bottleneck is also the same as RD4AD [8]. For each decoder layer, two 1x1 convolutions are used to project the feature map to reconstruct the frozen encoder and trained encoder, respectively. AdamW optimizer [33] is utilized with β=(0.9,0.999) and weight decay=1e-5. The learning rates of the new (decoder and bottleneck) and pre-trained (encoder) parameters are 2e-3 and 1e-5 respectively. The network is trained for 3,000 iterations on each sub-dataset in VisA, 2,000 on each sub-dataset in MVTec AD and ISIC2018, and 1,000 on APTOS and OCT2017. The batch size is 16 for industrial datasets and 32 for medical datasets. The α in (5) linearly rises from -3 to 1 in the first one-tenth iterations and keeps 1 for the rest of the training. By default, the batchnorm (BN) layers of encoder are set to *train* mode during training. Because training instability and performance drop are observed for some categories [1], the BN of encoder is set to *eval* mode (use pre-trained statistics) for such datasets. The reason is discussed in Appendix E. A library[2] is used to plot the landscape in Figure 4. The *distance* and *steps* arguments are set to 1 and 50 respectively.

On MVTec AD and VisA, the results of comparison methods are taken from the original papers or the benchmark paper [34] that report their performances. On medical datasets, we take the results from [30] if available; otherwise, we reproduce the method with official code. For RD4AD, we use their official code, in which $\mathcal{L}_{global}$ is already implemented instead of $\mathcal{L}_{region}$, as discussed in Section 2.1. Codes are implemented with Python 3.8 and PyTorch 1.12.0 cuda 11.3. Experiments are run on NVIDIA GeForce RTX3090 GPUs (24GB).

D Additional Experiments

D.1 Additional Experimental Results

In addition to the averaged segmentation performances in Table 2, we present the P-AUROC and AUPRO of each categories in MVTec AD in Table A1. The results in the main paper are reported

[1] *Toothbrush*, *Leather*, *Grid*, *Tile*, *Wood*, *Screw* in MVTec AD, *cashew*, *pcb1* in VisA, and OCT2017.

[2] https://github.com/marcellodebernardi/loss-landscapes

with a single random seed following our baseline [8]. In Table A2, we report the mean and standard deviation of three runs with three different random seeds (1, 11, and 111). In addition to the averaged performances in Table 3, we present the I-AUROC, P-AUROC, and AUPRO of each category in VisA in Table A3.

Table A1: Anomaly segmentation performance of 15 subset (categories) of MVTec AD (%).

	Carpet	Grid	Leather	Tile	Wood	Bottle	Cable	Capsule
P-AUROC	99.3	99.2	99.5	96.3	95.9	99.0	98.9	98.4
AUPRO	97.9	97.8	99.2	93.7	92.5	97.1	95.6	95.4

	Hazelnut	MetalNut	Pill	Screw	Toothbrush	Transistor	Zipper
P-AUROC	99.1	98.7	99.1	99.6	99.2	95.4	98.1
AUPRO	95.9	94.4	97.7	98.6	95.0	82.3	94.9

Table A2: Performance on MVTec AD over three runs (%).

	I-AUROC	P-AUROC	AUPRO
Carpet	99.72 ± 0.27	99.27 ± 0.05	98.55 ± 0.88
Grid	100 ± 0.00	99.26 ± 0.05	97.79 ± 0.03
Leather	100 ± 0.00	99.48 ± 0.02	99.17 ± 0.04
Tile	99.66 ± 0.17	96.18 ± 0.11	92.95 ± 0.66
Wood	99.05 ± 0.05	95.94 ± 0.03	92.61 ± 0.10
Bottle	100 ± 0.00	99.00 ± 0.01	97.13 ± 0.08
Cable	99.58 ± 0.15	98.92 ± 0.02	95.63 ± 0.02
Capsule	97.86 ± 0.36	98.40 ± 0.00	95.34 ± 0.04
Hazelnut	100 ± 0.00	99.08 ± 0.03	95.87 ± 0.08
MetalNut	100 ± 0.00	98.73 ± 0.04	94.49 ± 0.09
Pill	98.94 ± 0.24	99.13 ± 0.03	97.75 ± 0.05
Screw	97.80 ± 0.14	99.57 ± 0.02	98.50 ± 0.07
Toothbrush	99.44 ± 0.45	99.17 ± 0.02	94.99 ± 0.02
Transistor	99.65 ± 0.11	95.38 ± 0.03	82.61 ± 0.22
Zipper	99.68 ± 0.15	98.16 ± 0.04	95.12 ± 0.25
All Avg	99.42 ± 0.05	98.39 ± 0.01	95.21 ± 0.06

Table A3: Performance of 12 subset (categories) of VisA (%).

	candle	capsules	cashew	chewinggum	fryum	macaroni1
I-AUROC	97.20	93.55	98.14	99.28	97.56	98.84
P-AUROC	99.15	99.46	97.41	97.36	91.96	99.01
AUPRO	94.77	94.45	94.28	86.63	79.11	93.63

	macaroni2	pcb1	pcb2	pcb3	pcb4	pipe_fryum
I-AUROC	91.79	97.86	97.84	98.18	99.82	99.96
P-AUROC	99.04	99.79	98.99	99.05	98.70	98.30
AUPRO	97.36	96.76	92.47	95.14	91.29	95.57

D.2 Qualitative Visualization

The qualitative anomaly segmentation results of ReContrast trained by $\mathcal{L}_{global}$ and ReContrast trained by $\mathcal{L}_{global-hm}$ are presented in Figure A1. It is shown that the hard-normal regions are less activated with $\mathcal{L}_{global-hm}$. The results of VisA and medical images are shown in Figure A2 and Figure A3, respectively. Each $\mathcal{S}^{map}$ is min-max-normalized to 0-1 for clearer visualization.

D.3 Additional Ablation Study

Ablation study is conducted on the value of α in $\mathcal{L}_{global-hm}$, as shown in Table A4. Assuming the regional cosine distances $\mathcal{M}^k\ (h, w)$ conforms Gaussian distribution, the discarding rates of feature point are 2.3%, 15.9%, 50%, 69.1%, 84.1%, 93.3% and 97.7% for α = -2, -1, 0, 0.5, 1, 1.5 and 2, respectively. $\mathcal{L}_{global-hm}$ is not sensitive to the discarding rate within the range from 50% to 93%. Though α can be tuned for each dataset, we find $\alpha = 1$ works just well for most circumstances.

We test a variety of encoder backbones in Table A5, i.e., ResNet18, ResNet50, and WideResNet50 (default), and report their performances, model parameters, as well as multiply–accumulate operations (MACs). The corresponding decoder is the reversed version of the encoder. Different encoders may favor different training hyper-parameters such as learning rate and iteration. Though we do not further tune each backbone, all backbones produce excellent results with default hyper-parameters, suggesting the generality of our method. With each backbone, our method outperforms the corresponding feature reconstruction counterpart RD4AD [8]. Notably. our method with ResNet18 is comparable to RD4AD with WideResNet50.

The utilization of two encoders (one frozen, one trained domain-specific) in our method can function as an ensembling, which may also contribute to performances. We conduct ablation experiments using different encoder-decoder feature map pairs to generate anomaly maps in inference. The results are shown in Table A6. On MVTec, using only the feature of domain-specific encoder produces an I-AUROC of 99.38%, which is comparable to the 99.45% of the ReContrast default and outperforms the 98.86% of baseline Config. B. In addition, we train an ensembling version of Config. B using two different encoders (ResNet50 and WideResNet50), producing an I-AUROC of 98.94% which is nearly identical to the baseline. On APTOS, using only the feature of domain-specific encoder produces an I-AUROC of 97.73%, which outperforms both 97.51% of the ReContrast default and 92.49% of the baseline Config. B. The results indicate that the improvement due to ensembling is small and the domain-specific encoder is vital.

Table A4: Ablation on the values of α in $\mathcal{L}_{global-hm}$ on MVTec AD (%).

α	-inf ($\mathcal{L}_{global}$)	-2	-1	0	0.5	1	1.5	2
discard rate	0%	2.3%	15.9%	50%	69.1%	84.1%	93.3%	97.7%
I-AUROC	99.13	99.13	99.14	99.32	99.39	**99.45**	99.38	98.85
P-AUROC	98.09	98.11	98.13	98.30	98.31	98.37	**98.39**	98.25
AUPRO	94.59	94.62	94.65	94.97	95.00	95.20	**95.28**	95.13

Table A5: Ablation on the encoder backbones on MVTec AD (%).

Backbone	ResNet18		ResNet50		WResNet50	
Method	RD4AD	Ours	RD4AD	Ours	RD4AD	Ours
Params(M)	18.7	21.7	63.6	74.9	117	145
MACs(G)	4.95	10.1	19.4	42.0	36.0	75.2
I-AUROC	97.9	**98.7**	98.4	**99.1**	98.5	**99.5**
P-AUROC	97.1	**97.9**	97.7	**98.3**	97.8	**98.4**
AUPRO	91.2	**94.2**	93.1	**95.0**	93.9	**95.2**

Table A6: Ablation on the use of different encoder feature maps for calculating $\mathcal{S}^{map}$ (%).

	encoder feature maps for calculating $\mathcal{S}^{map}$	MVTec AD	APTOS
Config. B	3 frozen	98.86	92.49
Config. B (R50+WR50)	6 frozen	98.94	92.14
Ours	3 frozen only	99.16	95.32
Ours	3 trained only	99.38	**97.73**
Ours	3 trained + 3 frozen	**99.45**	97.51

E Limitations

Scope of Application. In this work, we mainly focus on UAD that detects regional defects (most common in practical applications), which is distinguished from one-class classification (OCC, or

Semantic AD). In our UAD, normal and anomalous samples are semantically the same objects except local detects, e.g. good cable v.s. spoiled cable. In OCC, normal samples and anomalous samples are semantically different, e.g. cat v.s. other animals. The slight difference in task setting makes the focus of corresponding methods different. Most methods for UAD attempt to detect anomalies by segmenting anomalous local regions, while methods for OCC detect anomalies based on the representation deviation of the whole image [35; 36]. Though methods of these two tasks can be used for each other to a certain extent, we mainly focus on UAD in this work.

Logical Anomaly. MVTec LOCO is a recently released dataset for evaluating the detection performance of logical anomalies. It contains 5 sub-datasets, each comprised of both structural anomalies, e.g. dents and scratches as in MVTec AD, and logical anomalies, e.g. dislocation and missing parts. We find our method performing less favorably on logical anomalous images than GCAD [37] which is specially designed for logical anomalies, as presented in Table A7. We still outperform other vanilla UAD methods that are not specially designed for such anomalies.

Table A7: Anomaly Detection Performance on MVTec LOCO (%). Structural I-AUROC is calculated on normal images and structural anomalous images. Logical I-AUROC is calculated on normal images and logical anomalous images. Mean I-AUROC is the average of the above two scores.

	RD4AD [8]	DRAEM [15]	PaDiM [17]	PatchCore [18]	GCAD [37]	Ours
struct. I-AUROC	88.0	74.4	70.5	82.0	80.6	**90.7**
logic. I-AUROC	69.4	72.8	63.7	69.0	**86.0**	73.4
mean I-AUROC	78.7	73.6	67.1	75.5	**83.3**	82.1

Batch Normalization and Training Instability. As discussed in Appendix C, training instability and performance drops are observed for a few categories when setting BN mode of the encoder to *train*. The phenomenon of training instability is observed in many deep learning tasks. However, validation sets are often allowed, which counteracts training instability and helps the choice of BN mode.

On the one hand, it can be caused by the limited feature diversity of the one-class UAD dataset. First, we recall the formulation of BN layer. Giving a d-dimensional feature $x = (x^{(1)}, ..., x^{(d)})$, each dimension (channel) is normalized in training:

$$\hat{x}^{(k)} = \frac{x^{(k)} - E\left[x^{(k)}\right]}{\sqrt{Var\left[x^{(k)}\right] + \epsilon}}$$

where the expectation E and variance Var are computed over the batch, and ϵ is a small constant for numerical stability. In most computer vision datasets, an input batch contains a variety of categories. Each dimension is more or less activated by different image features so that $Var\left[x^{(k)}\right]$ is not zero. However, because a UAD dataset contains only one type of category, there is a chance that a channel of pre-trained feature $x^{(k)}$ is barely activated or equally activated, e.g. a channel sensitive to animals is dead on industrial objects. Thus, the denominator $\sqrt{Var\left[x^{(k)}\right] + \epsilon}$ is nearly zero, causing malfunction of batch normalization. This problem can be fixed by using pre-trained statistical *running_mean* and *running_var* (*eval* mode), which loses the effect of BN.

On the other hand, this instability can be attributed to the feature of Adam optimizer. We found that a number of UAD methods [8; 21; 16] are also subject to training instability when running their code. Whether the last epoch (for reporting results) is in the middle of a loss spike and performance dip is strongly related to random seed. In some works [38; 39], the authors attribute the training instability to the historical estimation of squared gradients in Adam optimizer.

In our according explorations, we replace the batch variance $Var\left[x^{(k)}\right]$ smaller than min(5e-4, *running_var*) by pre-trained *running_var* and we reset the historical state buffer (gradient momentum and second-order gradient momentum) of Adam every 500 iters. Without manually selecting BN mode, such tricks enable training with more stability. It yields I-AUROC of 99.41% on MVTec and 97.28% on VisA, comparable to the 99.45% and 97.5% of our reported result and still outperforms previous SOTAs. In the future, approaches to eliminating the problem of BN and optimizer can be further investigated.

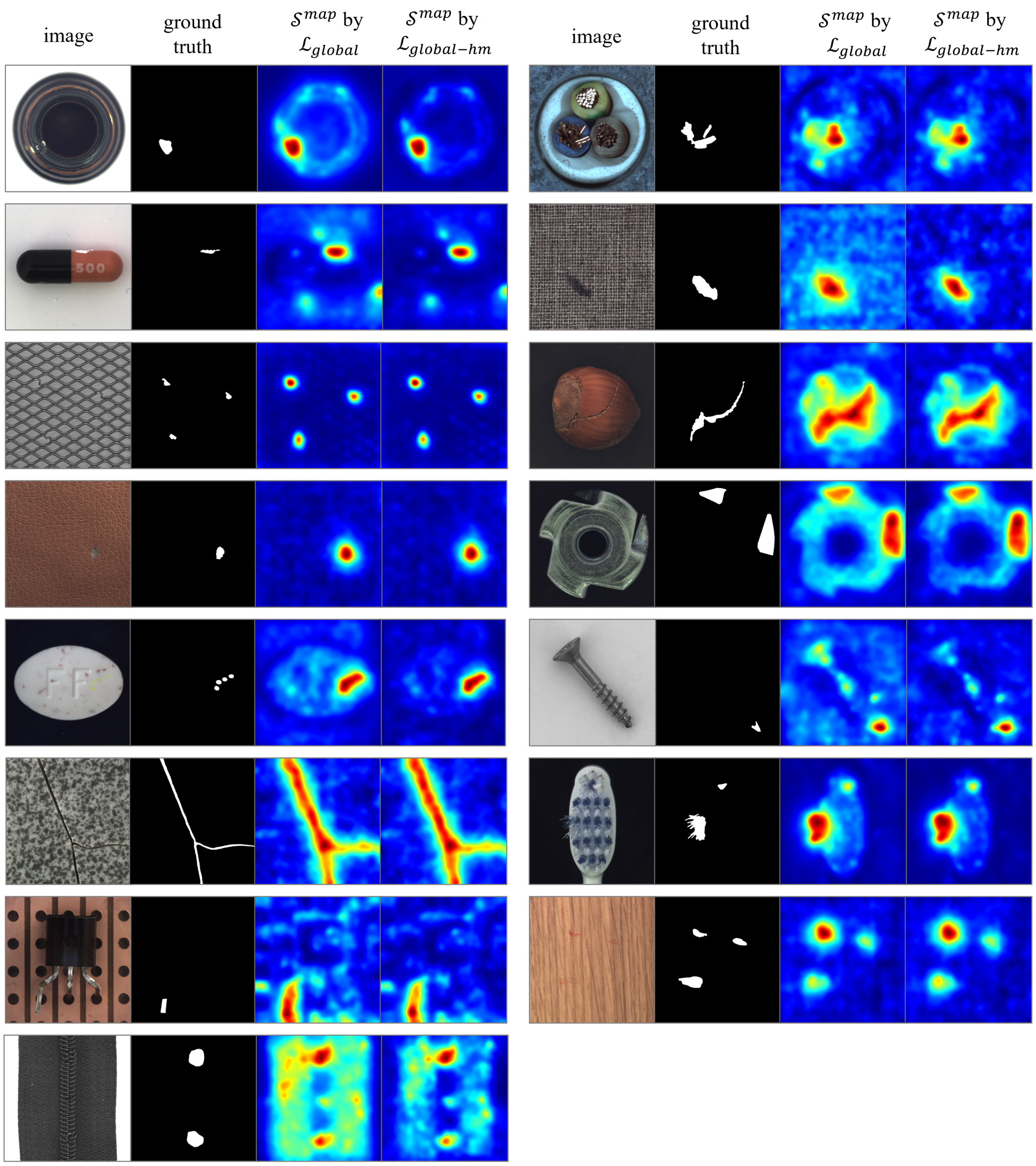
image
ground truth
$\mathcal{S}^{map}$ by $\mathcal{L}_{global}$
$\mathcal{S}^{map}$ by $\mathcal{L}_{global-hm}$
image
ground truth
$\mathcal{S}^{map}$ by $\mathcal{L}_{global}$
$\mathcal{S}^{map}$ by $\mathcal{L}_{global-hm}$
500

Figure A1: $\mathcal{S}^{map}$ of our ReContrast trained by $\mathcal{L}_{global}$ or $\mathcal{L}_{global-hm}$. Hard-normal regions are less activated with $\mathcal{L}_{global-hm}$.

References

[1] A. S. Collin and C. de Vleeschouwer, "Improved anomaly detection by training an autoencoder with skip connections on images corrupted with Stain-shaped noise," in *Proceedings - International Conference on Pattern Recognition*, 2020.

[2] P. Perera and V. M. Patel, "Learning Deep Features for One-Class Classification," *IEEE Transactions on Image Processing*, vol. 28, no. 11, 2019.

[3] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," in *2nd International Conference on Learning Representations, ICLR 2014 - Conference Track Proceedings*, 2014.

[4] F. Milkovic, B. Filipovic, M. Subasic, T. Petkovic, S. Loncaric, and M. Budimir, "Ultrasound anomaly detection based on variational autoencoders," in *International Symposium on Image and Signal Processing and Analysis, ISPA*, vol. 2021-September, 2021.

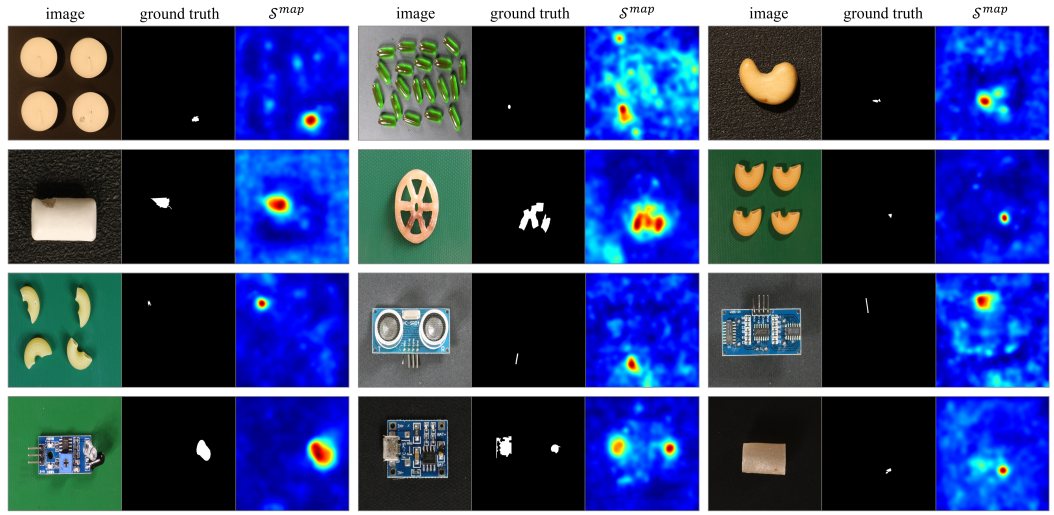

Figure A2: $\mathcal{S}^{map}$ of our method on VisA.

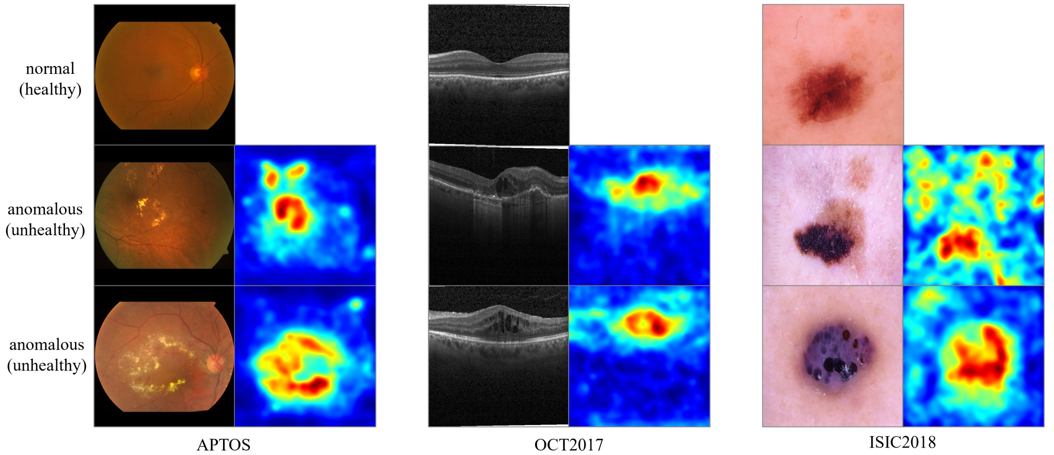

Figure A3: $\mathcal{S}^{map}$ of our method on medical image datasets.

[5] H. Zhao *et al.*, “Anomaly Detection for Medical Images Using Self-Supervised and Translation-Consistent Features,” *IEEE Transactions on Medical Imaging*, vol. 40, no. 12, 2021.

[6] T. Schlegl, P. Seeböck, S. M. Waldstein, G. Langs, and U. Schmidt-Erfurth, “f-AnoGAN: Fast unsupervised anomaly detection with generative adversarial networks,” *Medical Image Analysis*, vol. 54, 2019.

[7] S. Akcay, A. Atapour-Abarghouei, and T. P. Breckon, “GANomaly: Semi-supervised Anomaly Detection via Adversarial Training,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 11363 LNCS, 2019.

[8] H. Deng and X. Li, “Anomaly Detection via Reverse Distillation from One-Class Embedding,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9737–9746, 2022.

[9] Z. You *et al.*, “A unified model for multi-class anomaly detection,” in *Advances in Neural Information Processing Systems* (S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, eds.), vol. 35, pp. 4571–4584, Curran Associates, Inc., 2022.

[10] Z. You, K. Yang, W. Luo, L. Cui, Y. Zheng, and X. Le, “ADTR: Anomaly Detection Transformer with Feature Reconstruction,” *arXiv preprint arXiv:2209.01816*, 2022.

[11] Y. Shi, J. Yang, and Z. Qi, “Unsupervised anomaly segmentation via deep feature reconstruction,” *Neurocomputing*, vol. 424, 2021.

[12] M. Salehi, N. Sadjadi, S. Baselizadeh, M. H. Rohban, and H. R. Rabiee, “Multiresolution knowledge distillation for anomaly detection,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2021.

[13] G. Wang, S. Han, E. Ding, and D. Huang, “Student-teacher feature pyramid matching for anomaly detection,” in *The British Machine Vision Conference (BMVC)*, 2021.

[14] C. L. Li, K. Sohn, J. Yoon, and T. Pfister, “CutPaste: Self-Supervised Learning for Anomaly Detection and Localization,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2021.

[15] V. Zavrtanik, M. Kristan, and D. Skočaj, “DRÆM - A discriminatively trained reconstruction embedding for surface anomaly detection,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2021.

[16] Z. Liu, Y. Zhou, Y. Xu, and Z. Wang, “Simplenet: A simple network for image anomaly detection and localization,” *arXiv preprint arXiv:2303.15140*, 2023.

[17] T. Defard, A. Setkov, A. Loesch, and R. Audigier, “PaDiM: A Patch Distribution Modeling Framework for Anomaly Detection and Localization,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 12664 LNCS, 2021.

[18] K. Roth, L. Pemula, J. Zepeda, B. Schölkopf, T. Brox, and P. Gehler, “Towards total recall in industrial anomaly detection,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14318–14328, 2022.

[19] J. Yi and S. Yoon, “Patch SVDD: Patch-Level SVDD for Anomaly Detection and Segmentation,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 12627 LNCS, 2021.

[20] O. Rippel, P. Mertens, and D. Merhof, “Modeling the distribution of normal data in pre-trained deep features for anomaly detection,” in *Proceedings - International Conference on Pattern Recognition*, 2020.

[21] S. Lee, S. Lee, and B. C. Song, “Cfa: Coupled-hypersphere-based feature adaptation for target-oriented anomaly localization,” *IEEE Access*, vol. 10, pp. 78446–78454, 2022.

[22] T. Reiss, N. Cohen, L. Bergman, and Y. Hoshen, “Panda: Adapting pretrained features for anomaly detection and segmentation,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2806–2814, 2021.

[23] K. Sohn, C.-L. Li, J. Yoon, M. Jin, and T. Pfister, “Learning and evaluating representations for deep one-class classification,” in *International Conference on Learning Representations*, 2021.

[24] X. Gui, D. Wu, Y. Chang, and S. Fan, “Constrained adaptive projection with pretrained features for anomaly detection,” in *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22* (L. D. Raedt, ed.), pp. 2059–2065, International Joint Conferences on Artificial Intelligence Organization, 7 2022. Main Track.

[25] P. Bergmann, M. Fauser, D. Sattlegger, and C. 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*, pp. 9592–9600, 2019.

[26] Y. Zou, J. Jeong, L. Pemula, D. Zhang, and O. Dabeer, “Spot-the-difference self-supervised pre-training for anomaly detection and segmentation,” in *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXX*, pp. 392–408, Springer, 2022.

[27] D. S. Kermany *et al.*, "Identifying Medical Diagnoses and Treatable Diseases by Image-Based Deep Learning," *Cell*, vol. 172, no. 5, 2018.

[28] Asia Pacific Tele-Ophthalmology Society, "APTOS 2019 blindness detection," 2019.

[29] N. Codella *et al.*, "Skin lesion analysis toward melanoma detection 2018: A challenge hosted by the international skin imaging collaboration (isic)," *arXiv preprint arXiv:1902.03368*, 2019.

[30] Y. Zhao, Q. Ding, and X. Zhang, "AE-FLOW: Autoencoders with Normalizing Flows for Medical Images Anomaly Detection," in *The Eleventh International Conference on Learning Representations*, 2023.

[31] S. Zagoruyko and N. Komodakis, "Wide residual networks," *arXiv preprint arXiv:1605.07146*, 2016.

[32] O. Russakovsky *et al.*, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision*, vol. 115, no. 3, 2015.

[33] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *7th International Conference on Learning Representations, ICLR 2019*, 2019.

[34] G. Xie, J. Wang, J. Liu, J. Lyu, Y. Liu, C. Wang, F. Zheng, and Y. Jin, "Im-iad: Industrial image anomaly detection benchmark in manufacturing," *arXiv preprint arXiv:2301.13359*, 2023.

[35] B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson, "Estimating the support of a high-dimensional distribution," *Neural Computation*, vol. 13, no. 7, 2001.

[36] D. M. Tax and R. P. Duin, "Support Vector Data Description," *Machine Learning*, vol. 54, no. 1, 2004.

[37] P. Bergmann, K. Batzner, M. Fauser, D. Sattlegger, and C. Steger, "Beyond dents and scratches: Logical constraints in unsupervised anomaly detection and localization," *International Journal of Computer Vision*, vol. 130, no. 4, pp. 947–969, 2022.

[38] I. Molybog, P. Albert, M. Chen, Z. DeVito, D. Esiobu, N. Goyal, P. S. Koura, S. Narang, A. Poulton, R. Silva, *et al.*, "A theory on adam instability in large-scale machine learning," *arXiv preprint arXiv:2304.09871*, 2023.

[39] S. J. Reddi, S. Kale, and S. Kumar, "On the convergence of adam and beyond," *arXiv preprint arXiv:1904.09237*, 2019.