Siamese U-Net for Image Anomaly Detection and Segmentation with Contrastive Learning

Chia-Ying Lin[1]
lykaslin97@gapp.nthu.edu.tw
Shang-Hong Lai[1,2]
lai@cs.nthu.edu.tw

[1] Institute of Information Systems and Applications
National Tsing Hua University
Hsinchu, Taiwan

[2] Department of Computer Science
National Tsing Hua University
Hsinchu, Taiwan

Abstract

Computing image anomaly score from the maximum of the anomaly segmentation prediction result has been widely adopted for end-to-end anomaly detection approaches. However, slight discrepancy in predicted pixel-level anomaly scores for normal and anomalous features often leads to high segmentation accuracy but unmatched poor detection performance. To overcome this problem, we propose a novel siamese-based U-Net model based on a contrastive learning framework combined with deviation-based detection finetuning strategy. The model is trained to drag normal features together while alienating the anomaly samples. Moreover, we introduce a novel channel-positional attention module (CPAM) in our U-Net decoder for refined feature upsampling. Our model reaches SOTA performance on the well-known 2D MVTecAD dataset and outperforms all other methods on the challenging dataset MVTec3D-AD by a large margin.

1 Introduction

Anomaly detection and segmentation (AD&S) for industrial inspection is crucial in smart manufacturing as it enables precise and non-stop supervision far beyond human's capability. The primary target of the AD&S tasks is to determine whether the input sample belongs to the majority group or outliers of data points at either image- or pixel-level. With the mainstream for AD&S tasks [11][6][20][13][4][7] to train models under an unsupervised learning manner given defect-free data only, the normal-data-informed model is expected to generate different and distinguishable representations corresponding to the normal and anomalous data samples.

To allow end-to-end training and inference, existing methods tend to perform image-level detection based on the segmentation results, such as determining image-level anomaly based on the maximum of the anomaly map[4][13] or applying sliding windows on the segmentation map[17]. Although these unsupervised, segmentation-dominated approaches have reached excellent performance on the benchmark MVTecAD[8] dataset, they suffer from severe performance drop when applied to more challenging cases, such as MVTec3D-AD[1]. Two main issues regarding the performance drop are summarized as follows.

First, models trained in an unsupervised manner with normal data only are more likely to produce high false positive/negative rates as they have no access to true anomalies[9].

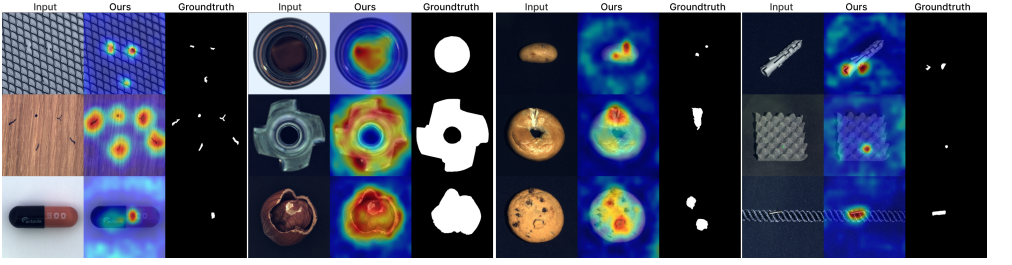

Figure 1: Visualization of the anomaly maps predicted by our model on MVTecAD[8](half-left) and MVTec3D-AD[1](half-right).

Secondly, segmentation-dominated models are prone to produce indistinguishable image-level anomaly scores for subtle anomalies. While the defects in MVTecAD[8] are mostly bird-eye detectable, MVTec3D-AD[1] contains certain defects which look nearly the same as normal from a single viewpoint, such as *raise* type in *cookie*. Slight discrepancy between pixel anomaly scores for normal and anomalous features often results in indiscriminative image-level anomaly scores, leading to poor anomaly detection performance.

To tackle the aforementioned issues, we propose a novel two-stage Siamese-UNet model trained with contrastive learning and deviation finetuning for the AD&S task. Our primary goal is to train a powerful U-Net decoder to extract representative and discriminative features for AD&S. We construct our basic U-Net block based on the T-S U-Net proposed in [4]. However, several improvements are introduced into our model and bring significant performance improvements over [4].

In terms of contrastive learning, instead of following an unsupervised learning setup as in [4], we adopt few anomalous samples to allow the model to effectively distinguish between positive and negative pairs from the feature learning. The model learns simultaneously to produce similar representations for the normal samples and push the anomalous samples away in a self-supervised fashion. Moreover, our proposed model is finetuned via a deviation manner by mapping normal and anomalous features numerically to a corresponding bi-polar scalar. Furthermore, we introduce a novel channel-positional attention fusion module, denoted as CPAM, which refines the feature extraction by applying attention mechanisms from the channel and spatial aspects.

Abundant experiments are conducted on various experimental setups, including full-shot, few-shot(16-shot) on benchmarks MVTecAD[8] and MVTec3D-AD[1] and also extended to RGB-D detection on MVTec3D-AD[1]. Our contributions can be summarized as follows:

- We propose a robust Siamese-UNet model with a two-stage training process, first with contrastive learning and followed by deviation finetuning, leading to more effective feature learning.
- We introduce a novel attention fusion module *CPAM*, which is proven to significantly refine features through the channel and spatial attentions.
- Our model provides SOTA accuracy on the benchmark MVTecAD[8] dataset and outperforms all the other methods on the challenging MVTec3D-AD[1] dataset by a large margin.

2 Related Works

2.1 Anomaly Detection and Segmentation

Unsupervised learning that trains models with defect-free samples is the most adopted technique for existing anomaly detection methods, including normalizing-flow(NF)-based [20][5][13][12], embedding-based[17][11][6], and reconstruction-based approaches [4][7].

Normalizing Flow (NF) is a series of bi-directional transformations that map input samples into specific probability distribution and thus performs precise likelihood estimation to detect out-of-distribution anomalies.[20][13][5][12] Embedding-based models expect a well-trained network to extract meaningful features representing input images or patches. The anomaly score can thus be calculated based on the embedding distance between the reference and the input sample. However, the high computational cost often hinders these approaches from being used in practice.[17][19]

On the other hand, variational autoencoder (VAE)[23], generative adversarial networks (GAN)[7][14], transformer[10], or U-Net[4] are widely adopted reconstruction-based approaches. Trained with normal samples, these methods are prone to fail in anomalous pixel restoration. Anomaly scores can therefore be determined based on pixel errors between groundtruth and reconstructed images. However, these models' high generalizability sometimes enables anomalous subregions reconstruction, resulting in inaccurate prediction.

Deng and Li [4] proposed a Teacher-Student U-Net framework, adopting reverse knowledge distillation with a Teacher encoder, Wide-ResNet50 pretrained on ImageNet, and a Student decoder. Unlike conventional reconstruction-based methods, this model[4] is trained to minimize each layer's feature-level difference between encoder and decoder, namely, to perform anomaly detection on the semantic feature space[4], which significantly improves the performance of the reconstruction-based approach.

However, the segmentation-dominated methods that determine image-level anomaly scores based on the maximum segmentation result share a common disadvantage: the model often produces a similar anomaly score for anomalies which appears similar to normal. In such conditions, slight discrepancy in predicted anomaly scores for normal and anomalous features often leads to high pixel-level accuracy yet unmatched poor detection result.

2.2 Self-Supervised Representation Learning

Recently, self-supervised learning (SSL) has shown its excellent capability in terms of representation learning without additional manual-labeled data. Tsai et al.[17] extended the SSL tasks of relative patch position prediction proposed in [19] into a more accurate angle direction of neighboring patches prediction, which is proven to improve model performance significantly. Moreover, the trained-from-scratch approach [15] proposed a novel SSL task, which first synthesizes anomalous samples via Poisson Image Editing and then trains the model to pinpoint those anomalous regions. The performance of [15] even surpasses several other methods that requires feature extractor for model training.

On the other hand, contrastive learning, targeting to drag normal features together while deviating from anomalous, is another potential yet ignored SSL approach. The Contrasting Shifted Instances (CSI) method [16] was proposed based on contrasting the original image with distribution-shifted augmented samples. The contrastive predictive coding [3] is a patch-level method focused on contrasting k patches in the same image with randomly matched N-1 negative samples. However, the SSL tasks mentioned above often learn from

defect-free samples only, which could encounter a performance drop or higher false positive rate for more challenging tasks.

3 Proposed Method

3.1 Model Architecture

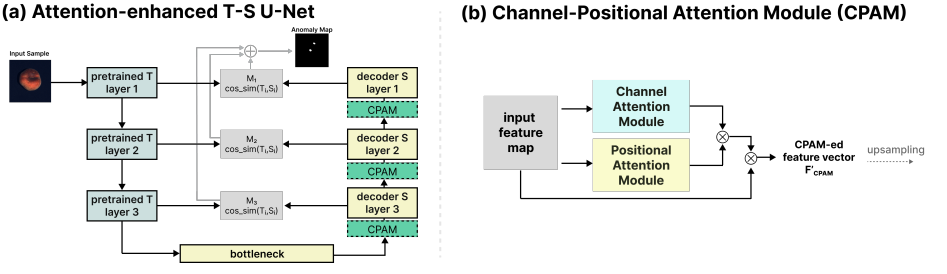

Figure 2: (a) Overview of the basic T-S U-Net block, embedded with channel-positional attention module (CPAM). Layerwise cosine similarity maps are aggregated as anomaly map. (b) CPAM comprises two attention submodules, applying attention from channel and spatial aspects.

3.1.1 Basic Block: Teacher-Student U-Net

Fig. 2 depicts the overview of the T-S U-Net architecture. Inspired by [4], we adopt the reverse distilled Teacher-Student U-Net (T-S U-Net) architecture as the basic block for our model. The T-S U-Net comprises a pretrained teacher encoder T and a student decoder S. For each layer in the U-Net structure, S learns to mimic the pretrained T layerwise behavior via restoring feature representation as similar as possible to the T's output embedding. Note that in this structure, T is frozen with no weight updates, whereas decoder S is optimized by the cosine similarity between the embedding from pretrained T and the restored feature vector from S with the following loss function:

$$L_{cossim} = \frac{1}{n}\sum_{i=1}^{n}(1 - cosine\_similarity(T_i, S_i)) \tag{1}$$

In eq.(1), T_i and S_i represent the i-th layer output from the pretrained Teacher encoder T and Student decoder S, respectively and n is the total number of layers in the U-Net structure. During the inference phase, the anomaly map for each input sample can be computed by:

$$M_{anomaly\_map} = \sum_{i=1}^{n}(1 - cosine\_similarity(T_i, S_i)) \tag{2}$$

To enable more robust feature restoration, we further embed an attention module, denoted as *CPAM* before every layer of the decoder S. Detailed description will be provided in the following section.

3.1.2 Channel-Positional Attention Module (CPAM)

To enable model learn from more robust feature representation, we refine the T-S U-Net via embedding a novel channel-positional attention module within each decoder block, denoted as *CPAM*. Fig.2 depicts the *CPAM* structure.

$$M_c = Q \otimes K, \quad F_c' = M_c \oplus V \tag{3}$$

$$M_c = Q \otimes K, \quad F_p' = M_p \oplus V \tag{4}$$

$$F_{CPAM}' = V \otimes F_c' \otimes F_p' \tag{5}$$

The channel and positional attentions are defined in eq.(3) and eq.(4), respectively. For channel attention, query, key, and value are first reshaped along the channel dimension, denoted as Q, K, and V. Hence, the inner-product from Q and K formulate the channel attention map M_c, where M_c and V are added as channel-attention refined feature vectors F_c'. On the other hand, we follow [22] for our positional attention mechanism, where Q, K, and V are first reshaped with a 2D convolution operation. Likewise, the product of Q and K generates corresponding positional attention map M_p and is added with original input vector as positional attention-guided feature vector F_p'. We multiply F_c', F_p', and the input feature vector V to derive the final CPAM-ed feature vectors for the following upsampling.

3.2 Model Flow: Two-Stage Training

Our model is trained in two consecutive stages. The first stage is a contrastive learning process designed specifically for the anomaly detection task. The second stage is a deviation-based detection finetuning process. Fig. 3 depicts an overview for each of these two stages for training the proposed Siamese U-Net model. The details of these two stages will be discussed in the subsequent subsections.

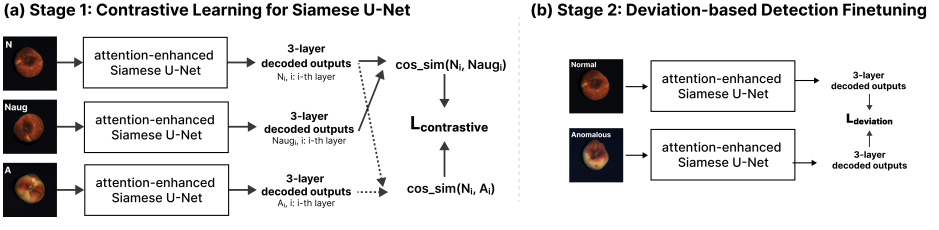

Figure 3: Overview of our proposed two-stage training flow.

3.2.1 Stage 1: Contrastive Learning

To improve model representation learning and generalizability on unseen anomalies, we employ contrastive learning in a self-supervised manner incorporated with the basic T-S U-Net block. Given a tuple of input sample consisting of a normal N, a random-rotated normal *Naug*, and an anomalous sample A, we triplicate the basic T-S U-Net block to construct a share-weight Siamese T-S U-Net. Note that *Naug* is the randomly rotated normal N, and

anomalous sample A is sampled from the testing set, which we ensure no overlaps between the training and testing set. In terms of contrastive learning, the model learns not only to output similar feature representation for normal samples but also to ensure the similarity between normal and anomalous samples is never higher than any of the normal pairs. To this end, we pair the outputs from the decoder with a positive and a negative pair. For positive pairs, we collect layer-wise restored features from N and $Naug$, whereas features from N and A are paired as negative pairs. The cosine similarity values for the positive and negative pairs are compared in the following contrastive loss:

$$L_{contrastive} = \frac{1}{n}\sum_{i=1}^{n} max(cos\_sim(N_i, A_i) - cos\_sim(N_i, Naug_i) + margin, 0), \tag{6}$$

where N_i, $Naug_i$, and A_i stand for the restored feature vectors from the i-th layer of the decoder for N, $Naug$ and A, respectively, and cos_sim refers to *cosine similarity*. An additional scalar *margin* ensures the similarity of the positive pair (N_i, $Naug_i$) is always larger than the negative (N_i, A_i) by certain *margin*. In combination with the L_{cossim}, the overall loss for stage-1 training can be formulated as:

$$L_{stage1} = \lambda_{con} L_{contrastive} + \lambda_{cos} L_{cossim} \tag{7}$$

3.2.2 Stage 2: Deviation-based Detection Finetuning

In the second stage, we finetune the model trained from stage-1 with a deviation strategy, namely, to map normal and anomalous features numerically to the corresponding bi-polar scalars to enhance the decoder with better representative feature restoration capability. Finetuning requires an input set consisting of one normal and one anomalous sample, denoted as (N, A). Note that the anomalous samples used at stage-2 are all sampled from the anomalies employed in stage-1. The layer-wise outputs from the decoder are retrieved for deviation, further optimized by the following deviation loss:

$$L_{deviation} = \frac{1}{n}\sum_{i=1}^{n}((1-y)\,|S_i| + y max(0, p - S_i)), \tag{8}$$

where y stands for the image-level groundtruth label (0: normal, 1: anomalous), S_i represents the i-th layer decoder output, and p represents the the bi-polar scalar for anomalous samples to be mapped to.

Fig. 4 illustrates how the anomaly synthesis module works. Following [15], the synthesis process is to first crop a small patch P_{src} from the source image X_{src} and then resize it randomly, denoted as P_{des}, and finally blend it on the destination image X_{des} to generate anomalous regions. TEXTURE and OBJECT synthesis for MVTecAD[3] differs slightly since for OBJECT, selected patch must ensure not to include background. Synthesis process is repeated several times to synthesize multiple types of anomalies. Note that image blending performed on the same image, namely, X_{src} is the same as X_{des}, is more likely to create too subtle defects[15]. We make sure that in our anomaly synthesis process, X_{src} and X_{des} are never duplicated.

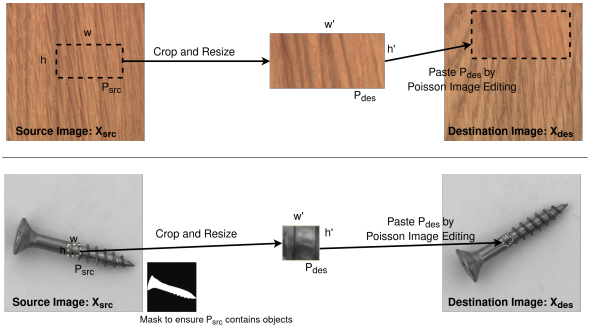

Figure 4: **Anomaly Synthesis Module.** The upper row depicts the synthesis steps for TEXTURE cases, whereas the lower row shows the synthesis steps for OBJECT classes based on the MVTecAD[8] dataset. As for MVTec3D-AD[1], anomalies are synthesized according to OBJECT settings. The anomaly synthesis module is based on [15].

4 Experiments

4.1 Datasets and Evaluation Metrics

We conduct our experiments mainly on the two industrial inspection benchmarks, MVTecAD [8] and MVTec3D-AD[1]. MVTecAD[8] consists of 5354 high-resolution RGB images from 15 real-world classes, with 3629 normal images for training and 1725 normal/anomalous images for testing. MVTec3D-AD[1] is a newly released 3D anomaly detection dataset, and it comprises over 4000 high-resolution 3D samples for 10 products, with each sample consisting of a RGB image, a set of organized point-cloud, and a 2D groundtruth. In our 2D experiment setup, we take the MVTec3D-AD[1] as a 2D dataset, leveraging its RGB data for model training without using the 3D geometric information. As our model training requires some anomalous samples for training, 10 anomalous samples from the original testing set for each product are randomly selected into the training set. Note that the selected anomalous samples are the same for the two-stage training, and we ensure no overlaps on the training and testing sets. We evaluate our model with image and pixel-level AUROC[8]. Moreover, since challenging cases usually contain subtle anomalies, where pixel-AUROC is often dominated by the rest of the normal features, we report pixel-PRO[2] as another performance evaluation, which is considered to be more informative for evaluating anomaly detection performance.

4.2 Experimental Comparison

Anomaly Scoring: As depicted in the attention-enhanced T-S U-Net basic block in Fig.2, we follow the way in [4] to obtain our anomaly segmentation results based on the upsampled feature differences between layer-wise T's output and the restored feature maps S. The maximum from the segmentation map is taken as the image-level anomaly score.

MVTecAD: Table 1 presents comparison of our method with other SOTAs on MVTecAD [8]. To allow fair analysis, we compare our method with both reconstruction-based approaches [4] and methods adopting SSL[21][18][3]. Our model surpasses all competitors and outperforms in most challenging classes, such as *transistor* and *screw*. Compared with

Table 1: Comparison on MVTecAD[8] with full-shot training setting. AUROC% is reported in the format of (image-level, pixel-level). Our results include two variants: trained with a few (10-shot) real anomalies and with synthetic data, denoted as Ours(r) and Ours(s), respectively.

	PatchCore	FastFlow	CS-Flow	Reverse Distillation	DRAEM+SSPCAB	RSTPM	CPC-AD	Ours(r)	Ours(s)
carpet	(-, -)	(**100.0, 99.4**)	(99.0, -)	(98.9, 97.0)	(98.2, 95.5)	(-, 99.0)	(80.9, -)	(**100.0**, 99.2)	(99.8, 99.3)
grid	(-, -)	(99.7, 98.3)	(100.0, -)	(100.0, 99.3)	(100.0, 99.7)	(-, 99.3)	(98.3, -)	(98.0, 99.3)	(**100.0**, 97.7)
leather	(-, -)	(100.0, 99.5)	(100.0, -)	(100.0, 99.4)	(100.0, 98.6)	(-, 99.0)	(99.0, -)	(98.0, 99.4)	(**100.0**, 99.5)
tile	(-, -)	(**100.0, 96.3**)	(100.0, -)	(99.3, 95.6)	(100.0, 99.2)	(-, 96.8)	(95.7, -)	(99.5, 95.9)	(99.1, 99.5)
wood	(-, -)	(**100.0, 97.0**)	(100.0, -)	(99.2, 95.3)	(99.5, 96.4)	(-, 96.4)	(80.3, -)	(**100.0**, 95.6)	(98.3, 95.7)
bottle	(-, -)	(100.0, 97.7)	(99.8, -)	(100.0, 98.7)	(98.4, 99.1)	(-, 98.9)	(99.8, -)	(**100.0**, 98.8)	(99.0, 98.6)
cable	(-, -)	(**100.0, 98.4**)	(97.1, -)	(95.0, 97.4)	(96.9, 94.7)	(-, 97.6)	(88.0, -)	(**100.0**, 97.7)	(96.8, 97.8)
capsule	(-, -)	(**100.0, 99.1**)	(98.6, -)	(96.3, 98.7)	(99.3, 94.3)	(-, 98.9)	(64.1, -)	(**100.0**, 98.9)	(97.6, 98.7)
hazelnut	(-, -)	(100.0, 99.1)	(99.3, -)	(99.9, 98.9)	(100.0, 99.7)	(-, 99.1)	(99.6, -)	(97.4, 99.1)	(**100.0**, 99.0)
metal_nut	(-, -)	(100.0, 98.5)	(99.7, -)	(100, 97.3)	(100.0, 99.5)	(-, 98.6)	(84.5, -)	(98.9, 97.7)	(**100.0**, 97.4)
pill	(-, -)	(**99.4, 99.2**)	(99.1, -)	(96.6, 98.2)	(99.8, 97.6)	(-, 97.1)	(92.1, -)	(99.6, 98.5)	(96.9, 98.0)
screw	(-, -)	(97.8, 99.4)	(99.6, -)	(97.0, 99.6)	(97.9, 97.6)	(-, 99.4)	(89.7, -)	(**100.0**, **99.7**)	(97.5, 99.4)
toothbrush	(-, -)	(94.4, 98.9)	(99.1, -)	(99.5, 99.1)	(100.0, 98.1)	(-, 99.0)	(87.8, -)	(98.0, **99.1**)	(99.2, 99.0)
transistor	(-, -)	(**99.8, 97.3**)	(97.6, -)	(96.7, 92.5)	(92.9, 90.9)	(-, 88.1)	(92.5, -)	(99.5, 93.2)	(97.1, 92.3)
zipper	(-, -)	(**99.5**, 98.7)	(91.9, -)	(98.5, 98.2)	(100.0, 98.8)	(-, 98.5)	(99.3, -)	(98.7, 98.8)	(97.8, 98.1)
Total AVG	(**99.6**, **98.4**)	(99.4, 98.5)	(98.7, -)	(98.5, 97.9)	(98.9, 97.3)	(96.9, 97.7)	(90.1, -)	(99.2, 98.1)	(98.6, 98.6)

Table 2: Comparison on MVTecAD[8] with few-shot(16-shot) setting. AUROC% is formatted as (image-level, pixel-level). Our model outperforms all listed approaches under extreme condition, setting a new SOTA record.

	CS-Flow	DifferNet	Reverse Distillation	DRAEM	NSA	Ours(r)
Total AVG	(93.8, -)	(83.6, -)	(91.0, 96.4)	(92.3, 86.2)	(90.7, 92.8)	(**95.1**, **97.3**)

current SOTA[11][20], we reaches competitive performance. Table 2 briefly shows the superior average accuracy of our method under the more extreme few-shot(16-shot) setup, perfectly demonstrating our proposed model's robustness and efficiency in feature representation learning under the few-shot setting.

MVTec3D-AD: To further assess the robustness and generalizability of our proposed method, we compare our model with other SOTA methods on the more challenging 3D dataset MVTec3D-AD[1] under a 2D experiment setup. Models are trained with RGB image data only without using the additional 3D geometric information provided in the dataset. Our model surpasses all competitors with over **16.7%** and **7.4%** performance improvements on the average image and pixel-level AUROC scores, respectively. Notably, for the challenging classes, specifically *cookie*, *foam*, and *potato*, in which the average image-AUROC from the SOTAs reaches only 57.2%, 74.3%, and 55%, respectively, our model provides consistently accurate results, achieving **83.2%**, **93.5%**, and **90.3%**, respectively, for these object classes.

4.3 Ablation Study

To provide more in-depth insights into our proposed framework, we conduct various ablation studies on MVTec3D-AD[1] to verify the impact of each component in our model.

Impact of Attention Fusion Module To enable precise anomaly segmentation, the quality of the restored feature for each decoder layer is the decisive factor. To this end, a common way for better feature representation learning is to refine features with an attention mechanism. Table 4 briefly quantifies the influence on the embedded attention module *CPAM* in our proposed method. On the top row of the left subtable of Table 4, we show that removing *CPAM* can directly lead to a significant performance drop in the overall AUROC score. Moreover, in the right subtable of Table 4, we further investigate the influence of fusion

Table 3: Quantitative comparison with SOTA methods on MVTec3D-AD[8] under 2D setup. AUROC% is formatted as (image-level, pixel-level). Our results include trained with a few (10-shot) real-world anomalies and with synthetic data, denoted as Ours(r) and Ours(s), respectively.

	PaDim	PatchCore	FastFlow	CFlow	DifferNet	CSFlow	Reverse Distillation	Ours(r)	Ours(s)
bagel	(97.5, 98.0)	(91.2, 89.9)	(89.3, 88.0)	(88.0, 85.5)	(81.9, -)	(89.4, -)	(96.2, 99.1)	**(99.2, 99.4)**	(99.0, 98.3)
cable gland	(77.5, 94.4)	(90.2, 95.3)	(62.0, 75.2)	(85.8, 91.9)	(67.0, -)	(91.7, -)	(88.7, **99.4**)	**(100.0, 99.4)**	(91.9, 97.8)
carrot	(69.8, 94.5)	(88.5, 95.7)	(79.5, 92.3)	(82.8, 95.8)	(61.2, -)	(74.9, -)	(94.2, **99.5**)	**(96.0, 99.5)**	(91.0, 98.9)
cookie	(58.2, 92.5)	(70.9, 91.8)	(42.6, 81.2)	(56.3, 86.7)	(48.4, -)	(66.8, -)	(62.4, 98.0)	**(83.2, 98.4)**	(65.8, 98.9)
dowel	(95.9, 96.1)	(95.2, 93.0)	(88.0, 92.9)	(98.6, 96.9)	(63.4, -)	(93.8, -)	(98.1, 99.5)	**(98.5, 99.6)**	(98.4, 99.2)
foam	(66.3, 79.2)	(73.3, 71.9)	(72.8, 64.6)	(73.8, 50.0)	(68.9, -)	(89.7, -)	(85.2, **96.7**)	**(93.5**, 94.9)	(83.1, 82.9)
peach	(85.8, 96.6)	(72.7, 92.0)	(65.1, 78.2)	(75.7, 88.9)	(65.5, -)	(60.3, -)	(**90.2**, **99.4**)	(81.3, **99.4**)	(89.0, 97.8)
Potato	(53.5, 94.0)	(56.2, 93.7)	(56.0, 61.5)	(62.8, 93.5)	(60.0, -)	(41.9, -)	(68.2, 99.2)	(**90.3, 99.3**)	(66.8, 98.3)
Rope	(83.2, 93.7)	(96.2, 93.8)	(98.2, 91.3)	(97.0, 90.4)	(72.9, -)	(97.1, -)	(**98.4**, 99.4)	(92.4, **99.6**)	(**98.8**, 96.6)
Tire	(76.0, 91.2)	(76.8, 92.9)	(61.3, 55.0)	(72.0, 91.9)	(53.6, -)	(72.6, -)	(73.0, 98.5)	**(98.6, 98.9)**	(81.2, 94.1)
AVG	(76.4, 93.0)	(81.1, 91.0)	(71.5, 78.0)	(79.3, 87.1)	(64.3, -)	(77.8, -)	(85.5, **98.8**)	(**93.3, 98.8**)	(86.5, 95.4)

module structure. We compare the performance with a different attention module variant (*CBAM*). The result indicates that our *CPAM* can reach superior performance on all evaluation metrics. Regarding image-level detection, *CPAM* outperforms *CBAM* by a 3.7% average AUROC score. In short, Table 4 justifies that the incorporation of attention module into our U-Net decoder can significantly improve the overall performance, and it also shows that the proposed *CPAM* is a more suitable attention module compared to another variant in this task.

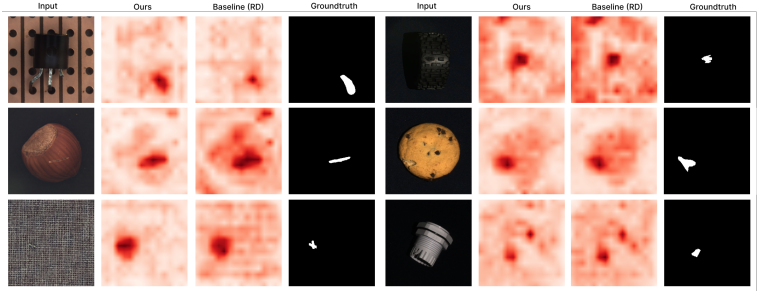

Figure 5: Qualitative comparison between our method and the baseline Reverse Distillation[4] on MVTecAD[8](left) and MVTec3D-AD[1](right).

Impact of Contrastive Learning: Table 4 shows the impact of contrastive learning in our method. To compare performance without the stage-1 contrastive learning module, we train our model with normal samples only, optimized by L_{cossim} in an unsupervised manner with a single basic U-Net block. The large performance drops for all evaluation metrics justify that the proposed SSL strategy is vital for the overall performance and provides better feature representation learning.

Impact of Deviation Finetuning: We also explore the importance of the stage-2 deviation-based finetuning in Table 4. Experiments indicate that performance of model without finetuning drops significantly, especially on the anomaly detection AUROC score, which further justifies our claim that pure segmentation-dominated approaches may fail to provide discriminative feature restoration for challenging cases, leading to a significant reduction in the image-level detection performance.

Table 4: Ablation study for our method on MVTec3D-AD[1]. The left sub-table justifies the impact of each component in our proposed framework, whereas the right sub-table further addresses on the influences regarding different attention fusion module structures. Image-, pixel-level AUROC, and pixel-level PRO are reported as *det*, *seg*, and *pro*, respectively.

	All Classes		
Task	det	seg	pro
Ours (w/o CPAM)	85.8	96.7	95.7
Ours (w/o finetune)	87.3	98.8	96.5
Ours (w/o Contrastive Learning)	88.7	93.3	95.2
Ours	93.3	98.8	98.4

	All Classes		
Task	det	seg	pro
Ours (CBAM)	83.6	98.7	96.0
Ours (CPAM)	87.3	98.8	96.5

4.4 Qualitative Results

We visualize the anomaly segmentation results for several classes from MVTecAD[8] and MVTec3D-AD[1] for qualitative comparison. Fig.5 shows that our proposed method can precisely localize and generate discriminative anomaly scores, regardless of the defects' shapes, sizes, or angles. For MVTecAD[8] results on the left half of Fig.5, our model localizes more precisely than the base *tire* on the top row of the right-half of Fig5, our model can still generate a reliable anomaly map due to the aid of the deviation finetuning. Without deviation finetuning, the baseline tends to falsely predict the background with a high anomaly score. Notably, even though several occluded anomalies in MVTec3D-AD[1] are hard to be detected with limited RGB information, the highlighted anomalous area still proves the robustness of our model of pinpointing those defects areas in all cases.

5 Conclusion

In this paper, we proposed a novel Siamese U-Net model trained with a two-stage learning strategy, i.e. contrastive learning and deviation finetuning. Our model includes a novel attention fusion module *CPAM* into U-Net to refine image representation learning for anomaly detection and segmentation. Extensive results demonstrate that our proposed method can not only achieve SOTA performance on the benchmark dataset, but also improve robustness and generalizability that bring outstanding performance on more challenging experiment setups, including few-shot (16-shot) on MVTec[8] and RGB anomaly detection and segmentation task on MVTec3D-AD[1].

References

[1] Paul Bergmann, Xin Jin, David Sattlegger, and Carsten Steger. The mvtec 3d-ad dataset for unsupervised 3d anomaly detection and localization. *arXiv preprint arXiv:2112.09045*, 2021.

[2] Andrew P. Bradley. The use of the area under the roc curve in the evaluation of machine learning algorithms. *Pattern Recognition*, 30(7):1145–1159, 1997. ISSN 0031-3203. doi: https://doi.org/10.1016/S0031-3203(96)00142-2. URL https://www.sciencedirect.com/science/article/pii/S0031320396001422.

[3] Puck de Haan and Sindy Löwe. Contrastive predictive coding for anomaly detection. *arXiv preprint arXiv:2107.07820*, 2021.

[4] 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.

[5] 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.

[6] Sungwook Lee, Seunghyun Lee, and Byung Cheol Song. Cfa: Coupled-hypersphere-based feature adaptation for target-oriented anomaly localization. *arXiv preprint arXiv:2206.04325*, 2022.

[7] Yufei Liang, Jiangning Zhang, Shiwei Zhao, Runze Wu, Yong Liu, and Shuwen Pan. Omni-frequency channel-selection representations for unsupervised anomaly detection. *arXiv preprint arXiv:2203.00259*, 2022.

[8] Bergmann P, Batzner K, Fauser M, Sattlegger D, and Steger C. The mvtec anomaly detection dataset: A comprehensive real-world dataset for unsupervised anomaly detection. *International Journal of Computer Vision*, abs/1904.02639, 2021.

[9] Guansong Pang, Choubo Ding, Chunhua Shen, and Anton van den Hengel. Explainable deep few-shot anomaly detection with deviation networks. *arXiv preprint arXiv:2108.00462*, 2021.

[10] Jonathan Pirnay and Keng Chai. Inpainting transformer for anomaly detection. In *International Conference on Image Analysis and Processing*, pages 394–406. Springer, 2022.

[11] 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.

[12] Marco Rudolph, Bastian Wandt, and Bodo Rosenhahn. Same same but differnet: Semi-supervised defect detection with normalizing flows. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 1907–1916, 2021.

[13] Marco Rudolph, Tom Wehrbein, Bodo Rosenhahn, and Bastian Wandt. Fully convolutional cross-scale-flows for image-based defect detection. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1088–1097, 2022.

[14] Thomas Schlegl, Philipp Seeböck, Sebastian M Waldstein, Ursula Schmidt-Erfurth, and Georg Langs. Unsupervised anomaly detection with generative adversarial networks to guide marker discovery. In *International conference on information processing in medical imaging*, pages 146–157. Springer, 2017.

[15] Hannah M Schlüter, Jeremy Tan, Benjamin Hou, and Bernhard Kainz. Self-supervised out-of-distribution detection and localization with natural synthetic anomalies (nsa). *arXiv preprint arXiv:2109.15222*, 2021.

[16] Jihoon Tack, Sangwoo Mo, Jongheon Jeong, and Jinwoo Shin. Csi: Novelty detection via contrastive learning on distributionally shifted instances. *Advances in neural information processing systems*, 33:11839–11852, 2020.

[17] Chin-Chia Tsai, Tsung-Hsuan Wu, and Shang-Hong Lai. Multi-scale patch-based representation learning for image anomaly detection and segmentation. In *2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 3065–3073, 2022. doi: 10.1109/WACV51458.2022.00312.

[18] Shinji Yamada and Kazuhiro Hotta. Reconstruction student with attention for student-teacher pyramid matching. *arXiv preprint arXiv:2111.15376*, 2021.

[19] Jihun Yi and Sungroh Yoon. Patch svdd: Patch-level svdd for anomaly detection and segmentation. In *Proceedings of the Asian Conference on Computer Vision*, 2020.

[20] Jiawei Yu, Ye Zheng, Xiang Wang, Wei Li, Yushuang Wu, Rui Zhao, and Liwei Wu. Fastflow: Unsupervised anomaly detection and localization via 2d normalizing flows. *arXiv preprint arXiv:2111.07677*, 2021.

[21] 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.

[22] Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. Self-attention generative adversarial networks. In *International conference on machine learning*, pages 7354–7363. PMLR, 2019.

[23] David Zimmerer, Fabian Isensee, Jens Petersen, Simon Kohl, and Klaus H. Maier-Hein. Unsupervised anomaly localization using variational auto-encoders. *CoRR*, abs/1907.02796, 2019. URL http://arxiv.org/abs/1907.02796.