Divide-and-Assemble: Learning Block-wise Memory for Unsupervised Anomaly Detection

Jinlei Hou[1,2†], Yingying Zhang[1], Qiaoyong Zhong[1], Di Xie[1], Shiliang Pu[1★], Hong Zhou[2]

[1]Hikvision Research Institute   [2]Zhejiang University

{houjinlei,zhangyingying7,zhongqiaoyong,xiedi,pushiliang.hri}@hikvision.com

zhouh@mail.bme.zju.edu.cn

Abstract

Reconstruction-based methods play an important role in unsupervised anomaly detection in images. Ideally, we expect a perfect reconstruction for normal samples and poor reconstruction for abnormal samples. Since the generalizability of deep neural networks is difficult to control, existing models such as autoencoder do not work well. In this work, we interpret the reconstruction of an image as a divide-and-assemble procedure. Surprisingly, by varying the granularity of division on feature maps, we are able to modulate the reconstruction capability of the model for both normal and abnormal samples. That is, finer granularity leads to better reconstruction, while coarser granularity leads to poorer reconstruction. With proper granularity, the gap between the reconstruction error of normal and abnormal samples can be maximized. The divide-and-assemble framework is implemented by embedding a novel multi-scale block-wise memory module into an autoencoder network. Besides, we introduce adversarial learning and explore the semantic latent representation of the discriminator, which improves the detection of subtle anomaly. We achieve state-of-the-art performance on the challenging MVTec AD dataset. Remarkably, we improve the vanilla autoencoder model by 10.1% in terms of the AUROC score.

1. Introduction

Anomaly detection in images is an increasingly important area among the computer vision community. Lacking of abnormal training data makes this task challenging and limits the use of well-established supervised learning methods. Hence, existing works define anomaly detection as an
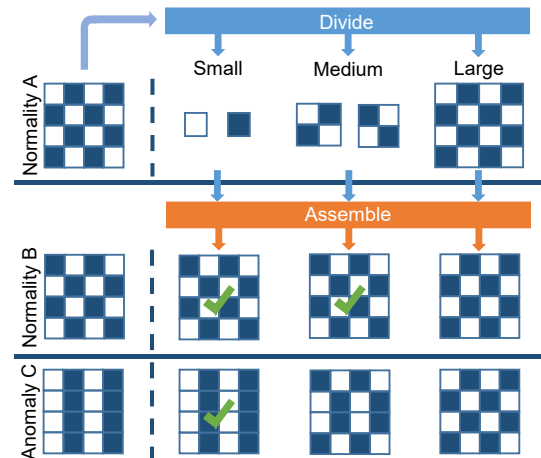
Figure 1. A toy example demonstrating that the reconstruction capability on normality and anomaly varies with different building block sizes. A: a normal sample with diagonal stripes. B: another normal sample with diagonal stripes similar to A. C: an abnormal sample with vertical stripes.

unsupervised learning problem that tries to model the distribution of normality without abnormal samples at training time. During inference, samples described as outliers of the normality distribution are then considered to be anomalous.

Reconstruction-based methods have been widely used for unsupervised anomaly detection. Most recent methods are based on convolutional neural networks (CNNs). In particular, autoencoder (AE) [21] is a natural choice to model the high-dimensional image data in the unsupervised setting. It is generally assumed [53, 17, 51] that the reconstruction error of the normal samples which are similar to the training data will be lower than abnormal samples. However, due to the existence of down-sampling in AE, the latent coding representation loses the detailed information of the original image. It may result in blurry output and large reconstruction error even for normal samples. We may strengthen the reconstruction capability of AE by reducing the down-sampling rate or adding skip connections [2].
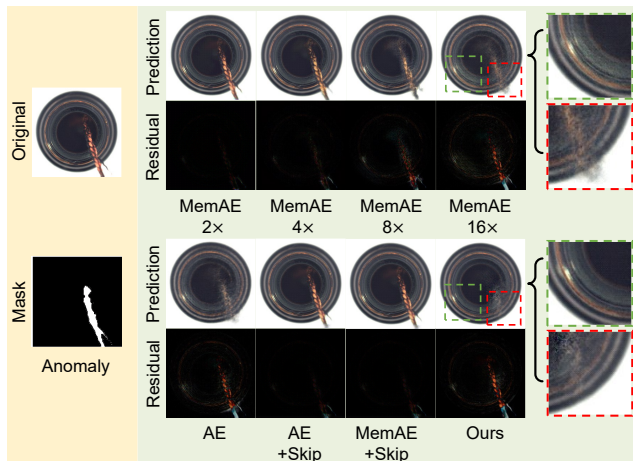
Figure 2. Exemplary reconstruction results of various models. **AE**: the vanilla convolutional autoencoder network whose encoder has a down-sampling rate of 16. **AE+Skip**: **AE** with skip connections. **MemAE**: **AE** augmented with the memory module proposed by [14] and $\{2\times, 4\times, 8\times, 16\times\}$ indicate the down-sampling rates in our reimplementation. **MemAE+Skip**: **AE+Skip** equipped with the memory module [14] in each skip connection.

But due to the uncontrollable generalizability of deep neural networks, it is difficult to improve the reconstruction quality of normal samples on one hand, and suppress the quality of abnormal samples on the other hand (see Figure 2). Thus, the vanilla AE does not work well.

To enlarge the gap of reconstruction error between the normal and abnormal samples, a few works [14, 28] have explored the memory module. In particular, an external memory module is utilized to explicitly model the distribution of normality. However, according to our investigation, the gap of reconstruction error can not be effectively enlarged in the existing works, leading to limited performance improvement. As shown in Figure 2, on the challenging MVTec AD [3] dataset, when the down-sampling rate is large (e.g. $16\times$), MemAE [14] failed to reconstruct both abnormal and some normal regions in the image. When the down-sampling rate is small (e.g. $2\times$) or with skip connections added, it reconstructed both normal and abnormal regions perfectly.

We argue that the major weakness of existing memory-augmented models lies in the fact that they decode the encoded feature maps in a per-pixel manner. If the basic building block of feature maps is as small as one pixel, then the variation of patterns of each block is limited. The smaller the block is, the more likely that anomaly shares the same block patterns with normality and thus can be reconstructed accurately. This hypothesis can be conceptually explained with a toy example shown in Figure 1. In this example, we try to learn block patterns of different sizes from normality A, and reconstruct normality B and anomaly C using the

learned blocks. When the block is small, we learn two simple patterns (i.e. white and dark blue), and succeed to reconstruct both normality B and anomaly C. When the block size is too large, we fail to reconstruct both B and C. With a proper medium block size, we are able to reconstruct perfectly for normality B and poorly for anomaly C, which is ideal for a reconstruction-based anomaly detection system.

Following the above motivation, we propose the Divide-and-Assemble Anomaly Detection (DAAD) framework. Specifically, we interpret the reconstruction of an image as a divide-and-assemble procedure. The encoded feature maps are divided evenly into a grid of blocks. Accordingly, a block-wise memory module which matches with the size of the division is introduced to model the block-level rather than pixel-level distribution of normal samples. This design makes it possible to modulate the reconstruction capability of the model. And thus the tradeoff between good reconstruction on normal samples and poor reconstruction on abnormal samples can be achieved. Furthermore, skip connections can be added safely to improve the reconstruction quality of normal samples without reconstruction of abnormal samples. Given an input image, DAAD uses the multi-scale features of multiple encoding layers as queries to retrieve the most relevant items in the corresponding memory banks. Then those items are aggregated and sent to the decoder through the corresponding skip connections. The memory items are learned end-to-end together with the model parameters via backpropagation. As shown in Figure 2, DAAD exhibits the desirable property of reconstructing details of normal regions while suppressing the reconstruction of abnormal regions.

In the real world, most anomalies such as scratch and crack tend to be subtle and occupy an extremely small region. Therefore even in abnormal images, the pixels are overwhelmed by normality rather than anomaly. For reconstruction-based methods, the cumulative error from normal pixels contributes more than abnormal pixels, which weakens the discriminativeness of the aggregated anomaly score. Even though the per-pixel reconstruction error of normal pixels is smaller than abnormal pixels, it is still difficult to detect anomaly. Thus we introduce an adversarially learned discriminator and exploit its low-dimensional semantic representation. The similarity between the features of the input image and the reconstructed image is treated as a measure of normality. The semantic feature-level anomaly score is complementary with the low-level reconstruction-based score, and their combination further improves the detection performance.

The major contributions of this paper are summarized as follows:

- We propose the multi-scale block-wise memory module which is able to achieve a tradeoff between good reconstruction on normality and poor reconstruction
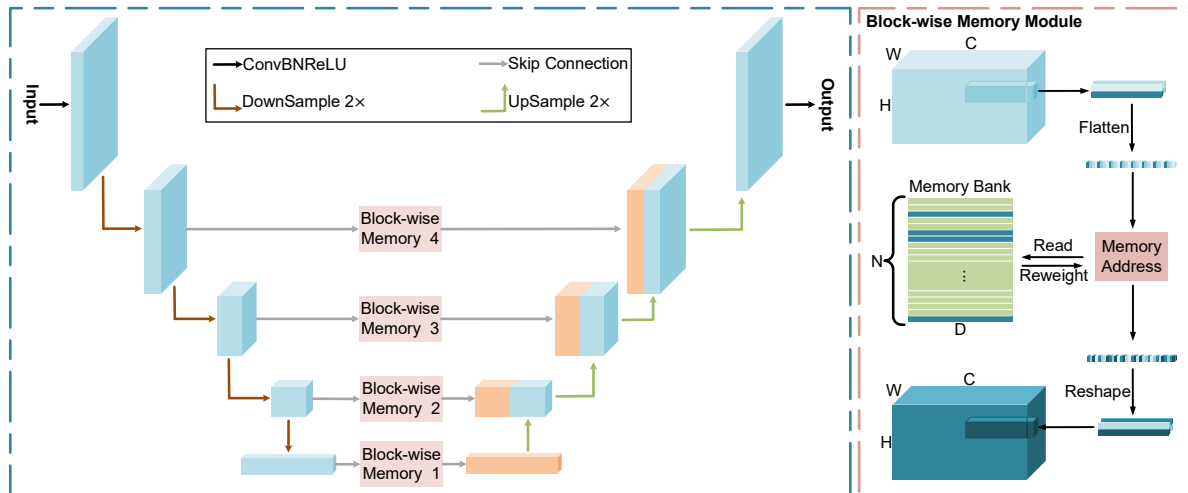
Figure 3. The architecture of the proposed DAAD framework. It consists of three major components: an encoder, a decoder and a group of block-wise memory modules. The encoder and decoder are connected with skip connections at multiple scales. And each skip connection is equipped with the block-wise memory module shown on the right.

on anomaly and thus make them well separable.

- We introduce an adversarially learned feature representation that detects anomaly in the semantic space and complements reconstruction-based detection.

- Extensive experiments on common unsupervised anomaly detection benchmarks demonstrate the excellent reconstruction and state-of-the-art performance.

2. Related Work

Unsupervised Learning for Anomaly Detection. Due to the scarcity and unpredictability of anomaly, anomaly detection is widely formulated as an unsupervised learning problem, where abnormal samples are unavailable during training [5]. Hence, the one-class classification methods such as one-class SVM [7], support vector data descriptor [40] and deep one-class networks [35, 4] are naturally chosen to solve this problem. Besides, unsupervised clustering methods (e.g. k-means method [52] and Gaussian Mixture Models [44]) have also been explored to model the behavior of normal samples in the latent space for separating anomaly from normality. However, these methods are usually found difficult to obtain effective feature representation, resulting in sub-optimal performance when faced with high-dimensional image data.

Reconstruction-based methods are mainly based on the assumption that a model trained only on normal samples can not represent and reconstruct anomaly accurately so that anomaly could be detected via the relatively higher reconstruction error than normality [53]. Sakurada et al. [37] applied autoencoder to anomaly detection for the first time. Later, Nicolau et al. [27] proposed to take low density in the hidden layer to indicate anomaly by combining autoencoder

and density estimation. To push the model to model the normality more effectively, recent works introduced more loss functions and network architectures. Sabokrou et al. [36] optimized the autoencoder with adversarial learning that could enhance the normal samples and distort the abnormal samples. Zenati et al. [49] trained a BiGAN model and employed an external encoder to learn the reverse process of the generator. Furthermore, Akcay et al. [2] encoded the input and the reconstruction to the low-dimensional latent space using the discriminator to calculate the reconstruction error. Similarly, Wang et al. [42] assumed that normal and abnormal data follow different Gaussian distributions and trained a variational autoencoder adversarially. Perera et al. [30] proposed the OCGAN that embeds two discriminators and a classifier on a denoising autoencoder. To suppress the reconstruction of anomaly, ITAE [19] and AESc [9] take the idea of image restoration. ITAE erases selected attributes (color or rotation) while AESc corrupts the original clean normality to simulate anomaly. Then they try to restore the corruptions in the training stage. Our work belongs to the reconstruction-based method. By extending the vanilla AE with the multi-scale block-wise memory module and the adversarially learned representation, we improve the baseline model significantly.

Feature-based methods exploit pre-trained deep representations in a transfer learning setting to model the distribution of normality. Recently, a few works such as SPADE [8], MAHA [33] and FAVAE [11] attempt to exploit the powerful knowledge learned from an external large-scale dataset. Specifically, a classification model is pretrained on the ImageNet dataset, which is subsequently used as a fixed feature extractor. Then SPADE and MAHA identify anomaly by comparing the semantic features of a test

sample with a set of normal samples. FAVAE uses the pre-trained model to guide the training of a variational auto-encoder. Our approach does not make use of any external knowledge, which makes the comparison unfair. Nevertheless, we compare their performance as a reference, and our approach still surpasses most feature-based methods.

Memory Networks. In order to capture long-term dependency in sequential data, recurrent neural network (RNN) and long short-term memory (LSTM) [18] were proposed. They use hidden states of the network to selectively record past information as their memory. But sometimes they fail to work well since the memory capacity is too small to accurately record all the contents of the sequential data. Recently, Weston et al. [43] introduced the memory networks that use a specialized memory bank that can be read and written and perform better memorization. However, it is hard to train the memory network via backpropagation due to the need of supervision for each layer during training. Sukhbaatar et al. [39] made it possible to train the memory networks in an end-to-end manner by using continuous memory representation to read/write the memory. Based on the end-to-end memory network, Miller et al. [26] utilized key-value pairs to further extend the memory capability. The success and effectiveness of the memory networks have attracted increasing interests for different computer vision tasks including object tracking [45, 23, 25], action recognition [48], visual question answering [47, 12] and anomaly detection [14, 28, 50, 46]. Similar to [14, 28], we also explore the potential of memory module in the context of anomaly detection. Our major new contribution is to modulate the reconstruction capability on normality and anomaly by generalizing the memory module to be block-wise rather than pixel-wise in a multi-scale fashion.

3. Approach

In this section, we first outline the proposed DAAD framework for unsupervised anomaly detection. Then, we present the detailed components of the block-wise memory module. Next, we introduce an adversarially learned feature representation that assists to identify anomaly by comparing the features of the input image and the reconstructed image. Finally, we describe the training loss function and the way to compute the anomaly score.

3.1. Network Architecture

3.1.1 Encoder and Decoder

The left part of Figure 3 shows the architecture of the proposed memory-augmented autoencoder network. It adopts a bow-tie network with an encoder G_E and a decoder G_D. The encoder network maps an input image x into a latent representation z. Being symmetric to G_E, the decoder network G_D upsamples the latent vector z back to the input

resolution and predicts the reconstruction, denoted as $\hat{x}$. Inspired by the UNet architecture [34] which is widely used for dense prediction tasks, we insert skip connections at multiple scales between the encoder and decoder. Using skip connections provides substantial advantages via direct information transfer between the layers. It can preserve both local and global information, and hence yield better reconstruction. However, it also brings a high risk of learning an identity mapping from input to output, which would make anomaly and normality inseparable as demonstrated in Figure 2. To address this issue, we equip all connections between the encoder and decoder with the block-wise memory module, which is described in the following section.

3.1.2 Block-wise Memory Module

The divide-and-assemble concept is effectively implemented using the block-wise memory module. It involves the block-wise query representation and a memory bank to record prototypical patterns. An attention-based addressing operator is designed for access and updating of the memory items.

Block-wise Query Representation. Given the input image x, the encoder emits a set of feature maps $\{f_1, f_2, \ldots, f_L\}$ where L is the number of scales. For each feature map f_l of size $H^l \times W^l \times C^l$, we divide it evenly into a grid of $Q = r_h \times r_w \times r_c$ blocks, where r_h, r_w and r_c are the division rates along height, width and channels respectively. The blocks in the grid are denoted as $\{\mathbf{q}^k \in \mathbb{R}^{\frac{H^l}{r_h} \times \frac{W^l}{r_w} \times \frac{C^l}{r_c}}, k = 1, 2, \ldots, Q\}$. Each block is flattened into a vector and serves as a query to the memory bank. For each query, unique features are aggregated from the memory bank with the memory addressing operation. The aggregated features share the same dimension as the query and are reshaped back into a block of $\frac{H^l}{r_h} \times \frac{W^l}{r_w} \times \frac{C^l}{r_c}$. By assembling the aggregated feature blocks, we obtain the transformed feature maps $\{\hat{f}_1, \hat{f}_2, \ldots, \hat{f}_L\}$, which are subsequently fed into the decoder G_D for the reconstruction of x. Note that we use the same set of division rates (r_h, r_w and r_c) for all scales, which results in different block sizes for feature maps of different levels. And the queries at different levels do not share the same memory bank.

Memory Bank and Addressing. The memory bank $\mathbf{M} \in \mathbb{R}^{N \times D}$ is defined as a real-valued matrix containing N memory items of fixed dimension D. It is used to explicitly record the prototypical normal patterns during training. To facilitate calculation, we set D equal to the dimension of the query $\mathbf{q}^k$. The i-th row vector of $\mathbf{M}$ denotes one memory item $\mathbf{m}_i \in \mathbb{R}^D$ ($i \in \{1, 2, \ldots, N\}$). Similar to [43, 32], the memory is content addressable which has a specific addressing scheme. It is addressed by computing the attention

weights $\mathbf{w}$ based on the similarity between the query $\mathbf{q}^k$ and each item $\mathbf{m}_i$ in the memory bank. Concretely, each weight w_i is computed as follows:

$$w_i = \frac{\exp\left(\frac{\mathbf{q}^k\mathbf{m}_i^\mathsf{T}}{\|\mathbf{q}^k\|\|\mathbf{m}_i\|}\right)}{\sum_{j=1}^{N}\exp\left(\frac{\mathbf{q}^k\mathbf{m}_j^\mathsf{T}}{\|\mathbf{q}^k\|\|\mathbf{m}_j\|}\right)} \qquad (1)$$

Note that w_i is non-negative and all elements in $\mathbf{w} \in \mathbb{R}^{1\times N}$ summate to one. Based on the attention weights $\mathbf{w}$, a new feature block $\hat{\mathbf{q}}^k$ is aggregated over the memory bank by

$$\hat{\mathbf{q}}^k = \mathbf{w}\mathbf{M} = \sum_{i=1}^{N} w_i \mathbf{m}_i \qquad (2)$$

As shown in Eq. (1) and (2), the addressing weights $\mathbf{w}$ is used to retrieve the memory items most related to $\mathbf{q}^k$ and generate the corresponding normal representation $\hat{\mathbf{q}}^k$. By linear combination over the memory items, some abnormal images may still achieve good reconstruction. To alleviate this problem, we drop the items whose memory weights are less than $\frac{1}{N}$ when computing $\hat{\mathbf{q}}^k$. Due to the sparsity in $\mathbf{w}$, only a small number of memory items can be addressed each time.

3.1.3 Adversarially Learned Representation

For some complex images that are difficult to reconstruct, the anomaly score based on the reconstruction loss easily suffers from the cumulative error of a large number of normal pixels. To address this issue, we introduce an adversarially learned representation. During training, we minimize the distance between the original image and the reconstructed image in the low-dimensional semantic representation space. During testing, we incorporate the feature similarity into the anomaly score to make it more robust in the case of poor reconstruction of normality. To prevent the learned features from mode collapse [16, 6], we utilize the adversarial training strategy. Specifically, the encoder-decoder network is treated as a generator G, and a discriminator D is appended after the decoder. As shown in Figure 4, the discriminator predicts the real/fake label of the given input. Its architecture is designed by following the discriminator of DCGAN [31]. During training, D tries to differentiate real images x from the fake ones $\hat{x}$ generated by G. During inference, the network D is used as a feature extractor to extract features of the input image x and the reconstructed image $\hat{x}$.

3.2. Loss Function

The overall training loss is a weighted summation of three losses:

$$\mathcal{L} = \lambda_{rec}\mathcal{L}_{rec} + \lambda_{adv}\mathcal{L}_{adv} + \lambda_{ali}\mathcal{L}_{ali} \qquad (3)$$
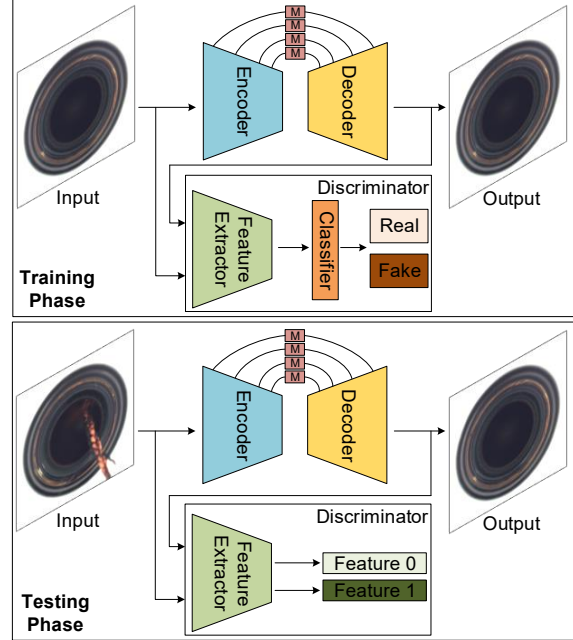

Figure 4. Diagram of the adversarially learned representation. A discriminator is appended after the decoder and the encoder-decoder network acts as the generator.

where $\mathcal{L}_{rec}$, $\mathcal{L}_{adv}$ and $\mathcal{L}_{ali}$ are the reconstruction, adversarial and alignment losses respectively. λ_{rec}, λ_{adv} and λ_{ali} are the corresponding loss weights.

Reconstruction Loss. The reconstruction loss makes the output image reconstructed from the generator G similar to its input image. Specifically, we apply L_2 distance between the input x and the reconstructed output $\hat{x}$, which ensures that our model is capable of reconstructing images contextually similar to normal images. The reconstruction loss is shown below:

$$\mathcal{L}_{\text{rec}} = \mathop{\mathbb{E}}_{x\sim p_x} \|x - \hat{x}\|_2 \qquad (4)$$

Here, p_x denotes the distribution of normal samples.

Adversarial Loss. To train the discriminator D for adversarially learned representation, we utilize the adversarial loss proposed in [15]. As shown in Eq. (5), the discriminator D tries to classify the real (input) and fake (reconstructed) samples correctly and the generator G tries to reconstruct x as realistically as possible. The adversarial loss $\mathcal{L}_{adv}$ can be written as:

$$\mathcal{L}_{adv} = \mathop{\mathbb{E}}_{x\sim p_x}\left[\log D(x)\right] + \mathop{\mathbb{E}}_{x\sim p_x}\left[\log(1 - D(G(x))\right] \qquad (5)$$

Alignment Loss. The alignment loss aims to match the feature representations of the reconstructed image $\hat{x}$ to the

	Category	Carpet	Grid	Leather	Tile	Wood	Bottle	Cable	Capsule	HN	MN	Pill	Screw	TB	TS	Zipper	Mean
	SPADE [8]	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0.855
Feature	MAHA [33]	1.000	0.897	1.000	0.998	0.996	1.000	0.950	0.951	0.991	0.947	0.887	0.852	0.969	0.955	0.979	**0.958**
	FAVAE [11]	0.671	0.970	0.675	0.805	0.948	0.999	0.950	0.804	0.993	0.852	0.821	0.837	0.958	0.932	0.972	0.879
	GeoTrans* [13]	0.437	0.619	0.841	0.417	0.611	0.744	0.783	0.670	0.359	0.813	0.630	0.500	0.972	0.869	0.820	0.672
	GANomaly* [1]	0.699	0.708	0.842	0.794	0.834	0.892	0.757	0.732	0.785	0.700	0.743	0.746	0.653	0.792	0.745	0.762
Recons.	ITAE [19]	0.706	0.883	0.862	0.735	0.923	0.941	0.832	0.681	0.855	0.667	0.786	1.000	1.000	0.843	0.876	0.839
	AESc [9]	0.89	0.97	0.89	0.99	0.95	0.98	0.89	0.74	0.94	0.73	0.84	0.74	1.00	0.91	0.94	0.89
	Ours	0.866	0.957	0.862	0.882	0.982	0.976	0.844	0.767	0.921	0.758	0.900	0.987	0.992	0.876	0.859	**0.895**

Table 1. Comparison with state-of-the-art feature-based and reconstruction-based methods on the MVTec AD dataset in terms of the image-level AUROC score. For clarity, the category names Hazelnut, Metal Nut, Toothbrush and Transistor are abbreviated as HN, MN, TB and TS respectively. * indicates reproduced results by [19].

input image x. It helps to ensure that the feature similarity is informative to indicate anomaly, i.e. higher for normal samples than abnormal samples. As shown in Figure 4, by feeding a pair of images (x and $\hat{x}$) into the discriminator D, we extract their features at the last convolutional layer of D. The extracted features $f_D(x)$ and $f_D(\hat{x})$ serve as semantic representations of x and $\hat{x}$. The alignment loss therefore is defined as follows:

$$\mathcal{L}_{ali} = \mathbb{E}_{x \sim p_x} ||f_D(x) - f_D(\hat{x})||_2 \quad (6)$$

3.3. Anomaly Score

During inference, anomaly is detected by fusing the reconstruction error in the high-dimensional image space and the alignment error in the low-dimensional feature space between the input image and the reconstructed image. Given the input image x, we define the anomaly score as follows:

$$\mathcal{A}(x) = \gamma R(x) + (1 - \gamma)L(x) \quad (7)$$

where $R(x)$ is the reconstruction score measuring the average pixel-wise difference between the input and the reconstructed images based on Eq. (4). $L(x)$ is the alignment score measured by the distance between the feature representations of the input and the reconstructed images based on Eq. (6). γ is a weighting parameter controlling the contribution of the reconstruction score and the alignment score. Note that both $R(x)$ and $L(x)$ are linearly scaled to the range of $[0, 1]$ before fusion.

4. Experiments

4.1. Setup

This section introduces the datasets, implementation details and the evaluation criterion used in our experiments.

Datasets. To demonstrate the proof of concept of the proposed approach, we validate the model on the challenging industrial anomaly detection dataset MVTec AD [3]. The MVTec AD dataset contains 5354 high-resolution color images of 15 different classes. 5 classes consist of textures such as wood or leather. The other 10 classes contain objects. Different from those used in the existing benchmarks, e.g. MNIST [24] and CIFAR10 [22], the anomalies in this dataset are more fine-grained and abnormal images come from the same category, which makes it more challenging. The dataset contains a training set with only normal images and a testing set with both normal and abnormal images.

Implementation Details. We resize each input image into a size of 256×256 for MVTec AD, and normalize the pixel values to the range of $[-1, 1]$. We use horizontal and vertical image flipping for data augmentation. Unless otherwise noted, r_h, r_w, r_c and the memory bank size N are empirically set to 8, 8, 1 and 500 to achieve the best overall performance. When only using the reconstruction loss (DAAD), the model is optimized via the Adam [20] optimizer with a fixed learning rate of $1e^{-4}$, a weight decay of 0, and momentums of $\beta_1 = 0.9$, $\beta_2 = 0.999$. If the adversarial loss and alignment loss are enabled (DAAD+), the learning rate is increased to $2e^{-4}$. The weighting parameters of $\mathcal{L}$ are empirically chosen as $\lambda_{rec} = 50$, $\lambda_{adv} = 0.5$ and $\lambda_{ali} = 0.5$. When applicable, the γ in Eq. (7) is set to 0.9. The model is implemented using PyTorch [29] and trained for 1000 epochs with a batch size of 8.

Evaluation. The performance of the model is evaluated by the area under the curve (AUC) of the receiver operating characteristics (ROC) at the image level (AUROC), a function of true positive rate over false positive rate with varying thresholds.

4.2. Results

MVTec AD. On this dataset, we compare our approach with the existing state-of-the-art methods, including both feature-based and reconstruction-based methods. As shown in Table 1, our approach surpasses most reconstruction-based methods by a large margin. Our performance is comparable to AESc [9] which exploits image restoration techniques by generating synthetic anomaly. We expect that these techniques are complementary to the proposed frame-

Method	MNIST	CIFAR10
VAE [21]	0.877	0.581
AnoGAN [38]	0.937	0.612
ADGAN [10]	0.947	0.624
GANomaly [1]	0.928	0.695
GeoTrans [13]	0.980	0.656
Skip-GANomaly [2]	-	0.732
OCGAN [30]	0.975	0.656
MemAE[1] [14]	0.975	0.609
CAVGA-D$_u$ [41]	0.986	0.737
Ours	**0.990**	**0.753**

[1] MemAE uses a different evaluation protocol, where only part of the test set are used.

Table 2. Comparison with existing methods on the MNIST and CIFAR10 datasets in terms of the AUROC score.

work. Notably, although the comparison is unfair due to their usage of external knowledge, we still outperform most feature-based methods, i.e. SPADE [8] and FAVAE [11]. Remarkably, compared with the vanilla AE, we obtain an absolute AUROC gain of 10.1% as discussed in Section 4.3.

MNIST and CIFAR10. To validate the versatility of the proposed framework, we report the performance on the commonly used MNIST and CIFAR10 datasets. We follow the one-class novelty detection protocol used in [30], and the comparison with existing reconstruction-based methods is listed in Table 2. Our approach also achieves excellent performance.

4.3. Ablation Study

To validate the effectiveness of the individual components of the proposed framework, we perform an extensive ablation study on the MVTec AD dataset.

Improvement over Baseline. We improve the vanilla AE and existing techniques like the memory module in [14] and skip connections in [2] significantly. The detailed comparison is shown in Table 3. AE is a vanilla autoencoder network trained with the reconstruction loss. MemAE is AE augmented with the pixel-wise memory module proposed in [14], which is re-implemented by using a single-scale (16×) block-wise memory module with $r_h = 16$, $r_w = 16$ and $r_c = 1$. AE+Skip is AE equipped with skip connections. DAAD is AE equipped with the proposed multi-scale block-wise memory module. DAAD+ is DAAD complemented with the adversarially learned representation. Taking AE as a baseline model, MemAE barely improves AE, indicating that the pixel-wise memory module does not work well for complex scenes like MVTec AD. AE+Skip fails to improve AE as it improves the reconstruction quality of both normality and abnormality, which makes them inseparable. We notice that AE+Skip can get

Category	AE	MemAE*	AE+Skip	DAAD	DAAD+
Carpet	0.411	0.454	0.385	0.671	0.866
Grid	0.841	0.946	0.879	0.975	0.957
Leather	0.615	0.611	0.570	0.628	0.862
Tile	0.696	0.630	0.986	0.825	0.882
Wood	0.961	0.967	0.977	0.957	0.982
Bottle	0.955	0.954	0.713	0.975	0.976
Cable	0.688	0.694	0.579	0.720	0.844
Capsule	0.819	0.831	0.747	0.866	0.767
Hazelnut	0.884	0.891	0.828	0.893	0.921
Metal Nut	0.565	0.537	0.336	0.552	0.758
Pill	0.882	0.883	0.853	0.898	0.900
Screw	0.956	0.992	1.000	1.000	0.987
Toothbrush	0.977	0.972	0.742	0.989	0.992
Transistor	0.776	0.793	0.749	0.814	0.876
Zipper	0.878	0.871	0.696	0.906	0.859
Mean	0.794	0.802	0.736	0.845	**0.895**

Table 3. AUROC improvement on MVTec AD over the vanilla AE and existing techniques. * denotes our reproduced results.

r_h & r_w	1	2	4	8	16
AUROC	0.716	0.784	0.821	**0.845**	0.813

Table 4. Impact of block size of the block-wise memory module on AUROC. Note that using a larger division rate results in a smaller block size.

good performance on some categories, e.g. tile, wood and screw. That is because the foreground and background texture of these categories are relatively homogeneous, which makes the reconstruction of anomaly more distinct. Nevertheless, compared with AE, DAAD achieves an AUROC gain of 5.1%, verifying the effectiveness of the proposed multi-scale block-wise memory module. DAAD+ further improves AUROC by 5%, resulting in a total improvement of 10.1%, which proves that the adversarially learned representation is complementary with DAAD.

Impact of Block Size. To validate our hypothesis on the impact of block size as demonstrated in Figure 1, we evaluate the AUROC score using different division rates along the spatial dimensions (r_h and r_w), which essentially determine the block size. As listed in Table 4, using either a small or large division rate leads to sub-optimal performance. While using a medium division rate (i.e. 8) achieves the best performance. In Figure 5, we visualize the reconstruction of two typical images under the same set of division rates. We can find that the reconstruction quality is closely related to the block size regulated by the division rate. For the object category of bottle, when the block size is large (with a small division rate), the model can only reconstruct the approximate appearance of the input and lose a lot of detailed texture, including the abnormal pixels. When the block size becomes smaller, the model can gradually
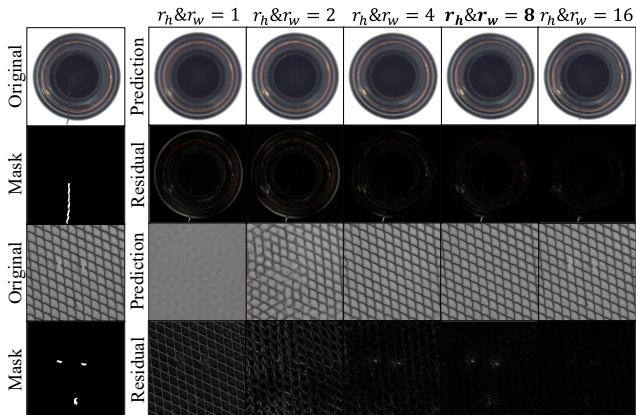
Figure 5. Two examples verifying that the reconstruction quality varies with the memory block size. The reconstructed images using a division rate (r_h & r_w) of 1, 2, 4, 8 and 16 are shown from left to right. As the division rate increases, the block size is smaller, and the reconstruction becomes more clear and even identical to the input image.

Method	$\mathcal{L}_{adv}$	$\mathcal{L}_{ali}$	$L(x)$	$R(x)$	AUROC
AE				✓	0.794
	✓			✓	0.812
	✓	✓		✓	0.817
	✓	✓	✓		0.819
	✓	✓	✓	✓	0.835
MemAE*				✓	0.802
	✓			✓	0.825
	✓	✓		✓	0.835
	✓	✓	✓		0.841
	✓	✓	✓	✓	0.853
DAAD				✓	0.845
	✓			✓	0.847
	✓	✓		✓	0.855
	✓	✓	✓		0.874
DAAD+	✓	✓	✓	✓	0.895

Table 5. Quantitative comparison for variants of adversarially learned representation on MVTec AD dataset.

reconstruct the details of the image. And when the block size is small enough, the model can even reconstruct abnormal pixels. For the texture category of grid, when the block size is too large, the model fails to reconstruct the original image. When the block size becomes smaller, the reconstructed texture gradually becomes complete and even identical to the input image. Similar results can be observed by varying the division rate along channels (r_c). In summary, our motivation of the divide-and-assemble procedure is well supported by both quantitative and qualitative results. By choosing a proper division rate, we are able to achieve a tradeoff between good reconstruction for normal pixels and poor reconstruction for abnormal pixels.
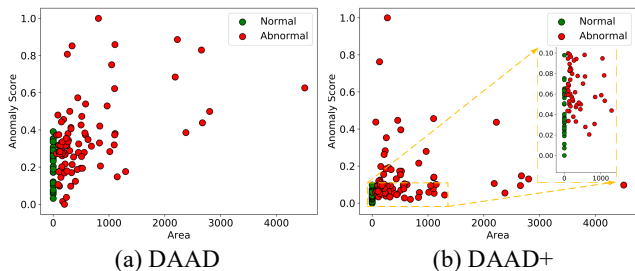

Figure 6. Scatter plot of the anomaly score and the anomaly area for the *leather* class of the MVTec AD dataset.

Impact of Adversarially Learned Representation. Table 5 shows the effectiveness of the adversarially learned representation based on different generators, i.e. AE, MemAE and DAAD. We observe that adversarial training alone brings no performance gain for DAAD, which indicates that DAAD has been able to achieve a perfect reconstruction, and it is difficult to further improve the performance based on the reconstruction score only by the adversarial loss $\mathcal{L}_{adv}$. The alignment loss $\mathcal{L}_{ali}$ essentially adds an additional constraint to the model, thereby achieving a small gain. However, we obtain a performance boost by integrating the alignment score $L(x)$ into the anomaly score. We believe that the reconstruction score $R(x)$ based on the input image and the reconstructed image can only attend to large or obvious anomalies. While the alignment score calculated in a low-dimensional space can further suppress the anomaly score of normality, especially for categories that are difficult to reconstruct, such as leather and carpet. Figure 6 compares the distribution of anomaly score of DAAD and DAAD+, which clearly validates our argument.

5. Conclusion

As a challenging unsupervised learning problem, many techniques have been exploited in the context of anomaly detection. The memory module perfectly fits the task of modeling the patterns of normality. Yet existing works failed to effectively improve the performance, especially on complex datasets like MVTec AD. In this work, we interpret the reconstruction of an image from the novel perspective of division and assembly of the feature maps. By controlling the granularity of the building blocks, i.e. the memory items, we achieve a balance between good reconstruction on normality and poor reconstruction on anomaly, which is the key to the success of reconstruction-based anomaly detection methods. Besides, using the adversarially learned feature representation, we are able to detect anomaly in the low-dimensional semantic space, which complements the detection in the image space. The excellent results on common anomaly detection benchmarks validate the effectiveness and versatility of the proposed framework.

References

[1] Samet Akcay, Amir Atapour-Abarghouei, and Toby P Breckon. Ganomaly: Semi-supervised anomaly detection via adversarial training. In *Asian conference on computer vision*, pages 622–637. Springer, 2018. 6, 7

[2] Samet Akçay, Amir Atapour-Abarghouei, and Toby P Breckon. Skip-ganomaly: Skip connected and adversarially trained encoder-decoder anomaly detection. In *2019 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2019. 1, 3, 7

[3] 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 Conference on Computer Vision and Pattern Recognition*, pages 9592–9600, 2019. 2, 6

[4] Raghavendra Chalapathy, Aditya Krishna Menon, and Sanjay Chawla. Anomaly detection using one-class neural networks. *arXiv preprint arXiv:1802.06360*, 2018. 3

[5] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM computing surveys (CSUR)*, 41(3):1–58, 2009. 3

[6] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. *arXiv preprint arXiv:2011.10566*, 2020. 5

[7] Yunqiang Chen, Xiang Sean Zhou, and Thomas S Huang. One-class svm for learning in image retrieval. In *Proceedings 2001 International Conference on Image Processing (Cat. No. 01CH37205)*, volume 1, pages 34–37. IEEE, 2001. 3

[8] Niv Cohen and Yedid Hoshen. Sub-image anomaly detection with deep pyramid correspondences. *arXiv preprint arXiv:2005.02357*, 2020. 3, 6, 7

[9] Anne-Sophie Collin and Christophe De Vleeschouwer. Improved anomaly detection by training an autoencoder with skip connections on images corrupted with stain-shaped noise. *arXiv preprint arXiv:2008.12977*, 2020. 3, 6

[10] Lucas Deecke, Robert Vandermeulen, Lukas Ruff, Stephan Mandt, and Marius Kloft. Image anomaly detection with generative adversarial networks. In *Joint european conference on machine learning and knowledge discovery in databases*, pages 3–17. Springer, 2018. 7

[11] David Dehaene and Pierre Eline. Anomaly localization by modeling perceptual features. *arXiv preprint arXiv:2008.05369*, 2020. 3, 6, 7

[12] Chenyou Fan, Xiaofan Zhang, Shu Zhang, Wensheng Wang, Chi Zhang, and Heng Huang. Heterogeneous memory enhanced multimodal attention model for video question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1999–2007, 2019. 4

[13] Izhak Golan and Ran El-Yaniv. Deep anomaly detection using geometric transformations. In *Advances in Neural Information Processing Systems*, pages 9758–9769, 2018. 6, 7

[14] Dong Gong, Lingqiao Liu, Vuong Le, Budhaditya Saha, Moussa Reda Mansour, Svetha Venkatesh, and Anton van den Hengel. Memorizing normality to detect anomaly:

Memory-augmented deep autoencoder for unsupervised anomaly detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1705–1714, 2019. 2, 4, 7

[15] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014. 5

[16] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent: A new approach to self-supervised learning. *arXiv preprint arXiv:2006.07733*, 2020. 5

[17] Mahmudul Hasan, Jonghyun Choi, Jan Neumann, Amit K Roy-Chowdhury, and Larry S Davis. Learning temporal regularity in video sequences. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 733–742, 2016. 1

[18] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997. 4

[19] Chaoqing Huang, Jinkun Cao, Fei Ye, Maosen Li, Ya Zhang, and Cewu Lu. Inverse-transform autoencoder for anomaly detection. *arXiv preprint arXiv:1911.10676*, 2019. 3, 6

[20] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6

[21] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 1, 7

[22] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 6

[23] Zihang Lai, Erika Lu, and Weidi Xie. Mast: A memory-augmented self-supervised tracker. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6479–6488, 2020. 4

[24] Yann LeCun. The mnist database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*, 1998. 6

[25] Francesco Marchetti, Federico Becattini, Lorenzo Seidenari, and Alberto Del Bimbo. Mantra: Memory augmented networks for multiple trajectory prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7143–7152, 2020. 4

[26] Alexander Miller, Adam Fisch, Jesse Dodge, Amir-Hossein Karimi, Antoine Bordes, and Jason Weston. Key-value memory networks for directly reading documents. *arXiv preprint arXiv:1606.03126*, 2016. 4

[27] Miguel Nicolau, James McDermott, et al. A hybrid autoencoder and density estimation model for anomaly detection. In *International Conference on Parallel Problem Solving from Nature*, pages 717–726. Springer, 2016. 3

[28] Hyunjong Park, Jongyoun Noh, and Bumsub Ham. Learning memory-guided normality for anomaly detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14372–14381, 2020. 2, 4

[29] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban

Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in PyTorch. In *NIPS Autodiff Workshop*, 2017. 6

[30] Pramuditha Perera, Ramesh Nallapati, and Bing Xiang. Ocgan: One-class novelty detection using gans with constrained latent representations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2898–2906, 2019. 3, 7

[31] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015. 5

[32] Jack Rae, Jonathan J Hunt, Ivo Danihelka, Timothy Harley, Andrew W Senior, Gregory Wayne, Alex Graves, and Timothy Lillicrap. Scaling memory-augmented neural networks with sparse reads and writes. In *Advances in Neural Information Processing Systems*, pages 3621–3629, 2016. 4

[33] Oliver Rippel, Patrick Mertens, and Dorit Merhof. Modeling the distribution of normal data in pre-trained deep features for anomaly detection. *arXiv preprint arXiv:2005.14140*, 2020. 3, 6

[34] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015. 4

[35] Lukas Ruff, Robert Vandermeulen, Nico Goernitz, Lucas Deecke, Shoaib Ahmed Siddiqui, Alexander Binder, Emmanuel Müller, and Marius Kloft. Deep one-class classification. In *International conference on machine learning*, pages 4393–4402, 2018. 3

[36] Mohammad Sabokrou, Mohammad Khalooei, Mahmood Fathy, and Ehsan Adeli. Adversarially learned one-class classifier for novelty detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3379–3388, 2018. 3

[37] Mayu Sakurada and Takehisa Yairi. Anomaly detection using autoencoders with nonlinear dimensionality reduction. In *Proceedings of the MLSDA 2014 2nd Workshop on Machine Learning for Sensory Data Analysis*, pages 4–11, 2014. 3

[38] 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. 7

[39] Sainbayar Sukhbaatar, Jason Weston, Rob Fergus, et al. End-to-end memory networks. In *Advances in neural information processing systems*, pages 2440–2448, 2015. 4

[40] David MJ Tax and Robert PW Duin. Support vector data description. *Machine learning*, 54(1):45–66, 2004. 3

[41] Shashanka Venkataramanan, Kuan-Chuan Peng, Rajat Vikram Singh, and Abhijit Mahalanobis. Attention guided anomaly localization in images. In *Proceedings of the European conference on computer vision (ECCV)*, pages 485–503, 2020. 7

[42] Xuhong Wang, Ying Du, Shijie Lin, Ping Cui, Yuntian Shen, and Yupu Yang. advae: A self-adversarial variational autoencoder with gaussian anomaly prior knowledge for anomaly detection. *Knowledge-Based Systems*, 190:105187, 2020. 3

[43] Jason Weston, Sumit Chopra, and Antoine Bordes. Memory networks. *arXiv preprint arXiv:1410.3916*, 2014. 4

[44] Liang Xiong, Barnabás Póczos, and Jeff G Schneider. Group anomaly detection using flexible genre models. In *Advances in neural information processing systems*, pages 1071–1079, 2011. 3

[45] Tianyu Yang and Antoni B Chan. Learning dynamic memory networks for object tracking. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 152–167, 2018. 4

[46] Ziyi Yang, Teng Zhang, Iman Soltani Bozchalooi, and Eric Darve. Memory augmented generative adversarial networks for anomaly detection. *arXiv preprint arXiv:2002.02669*, 2020. 4

[47] Chengxiang Yin, Jian Tang, Zhiyuan Xu, and Yanzhi Wang. Memory augmented deep recurrent neural network for video question answering. *IEEE transactions on neural networks and learning systems*, 2019. 4

[48] Yuan Yuan, Dong Wang, and Qi Wang. Memory-augmented temporal dynamic learning for action recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 9167–9175, 2019. 4

[49] Houssam Zenati, Chuan Sheng Foo, Bruno Lecouat, Gaurav Manek, and Vijay Ramaseshan Chandrasekhar. Efficient gan-based anomaly detection. *arXiv preprint arXiv:1802.06222*, 2018. 3

[50] Chengfen Zhang, Yue Wang, Xinyu Zhao, Yan Guo, Guotong Xie, Chuanfeng Lv, and Bin Lv. Memory-augmented anomaly generative adversarial network for retinal oct images screening. In *2020 IEEE 17th International Symposium on Biomedical Imaging (ISBI)*, pages 1971–1974. IEEE, 2020. 4

[51] Yiru Zhao, Bing Deng, Chen Shen, Yao Liu, Hongtao Lu, and Xian-Sheng Hua. Spatio-temporal autoencoder for video anomaly detection. In *Proceedings of the 25th ACM international conference on Multimedia*, pages 1933–1941, 2017. 1

[52] Arthur Zimek, Erich Schubert, and Hans-Peter Kriegel. A survey on unsupervised outlier detection in high-dimensional numerical data. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, 5(5):363–387, 2012. 3

[53] Bo Zong, Qi Song, Martin Renqiang Min, Wei Cheng, Cristian Lumezanu, Daeki Cho, and Haifeng Chen. Deep autoencoding gaussian mixture model for unsupervised anomaly detection. In *International Conference on Learning Representations*, 2018. 1, 3