Context Enhancement with Reconstruction as Sequence for Unified Unsupervised Anomaly Detection

Hui-Yue Yang[a,b,c], **Hui Chen**[a,b], **Lihao Liu**[a,b], **Zijia Lin**[a], **Kai Chen**[a,b], **Liejun Wang**[d], **Jungong Han**[a,b] and **Guiguang Ding**[a,b,*]

[a]Tsinghua University
[b]BNRist
[c]Hangzhou Zhuoxi Institute of Brain and Intelligence
[d]Xinjiang University

Abstract. Unsupervised anomaly detection (AD) aims to train robust detection models using only normal samples, while can generalize well to unseen anomalies. Recent research focuses on a unified unsupervised AD setting in which only one model is trained for all classes, *i.e.,* n-class-one-model paradigm. Feature-reconstruction-based methods achieve state-of-the-art performance in this scenario. However, existing methods often suffer from a lack of sufficient contextual awareness, thereby compromising the quality of the reconstruction. To address this issue, we introduce a novel Reconstruction as Sequence (RAS) method, which enhances the contextual correspondence during feature reconstruction from a sequence modeling perspective. In particular, based on the transformer technique, we integrate a specialized RASFormer block into RAS. This block enables the capture of spatial relationships among different image regions and enhances sequential dependencies throughout the reconstruction process. By incorporating the RASFormer block, our RAS method achieves superior contextual awareness capabilities, leading to remarkable performance. Experimental results show that our RAS significantly outperforms competing methods, well demonstrating the effectiveness and superiority of our method. Our code is available at https://github.com/Nothingtolose9979/RAS

1 Introduction

Anomaly detection (AD) aims to identify outliers or abnormal regions for an input image. It is widely used in various fields such as industrial manufacturing [4, 43, 41, 26], healthcare [16, 35, 23], surveillance [28, 10, 36] and fraud detection[20, 7, 15, 22]. Developing optimal AD models is challenging due to the rarity of anomalies in real-world scenarios. Researchers have explored unsupervised anomaly detection without requiring anomaly-specific data. Nonetheless, they often build **separate** models for each class, *i.e.,* the n-class-n-model paradigm shown in Fig. 1 (left). However, due to the diversity of anomaly classes, such a paradigm may not be the best solution, especially as the number of classes increases [42].

Recently, developing a robust AD framework that achieves **unified** unsupervised anomaly detection has gained much attention [39, 42]. Such a unified setting can detect different anomalies for all classes with only one AD model, *i.e.,* n-class-one-model paradigm shown in
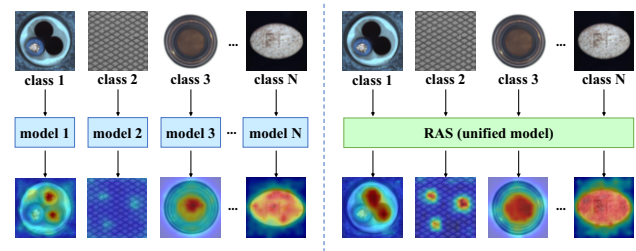

Figure 1. Comparison of different paradigms in anomaly detection. **Left:** n-class-n-model paradigm, where **separate** models are trained for each class. **Right:** n-class-one-model paradigm, utilizing a **unified** model to detect anomalies across all classes.

Fig. 1 (right). In this scenario, feature-reconstruction-based method has emerged as a popular method, owing to its simplicity, impressive detection performance, and robustness. These techniques focus on reconstructing the visual features of an input image during the feature reconstruction process, with anomaly regions identified by comparing the original image feature and the reconstructed one. For example, UniAD [39] first incorporates the transformer architecture with feature jittering and neighbor masked attention to amplify feature differences and improve the accuracy of anomaly detection. UniCon-HA [34] proposes unilaterally aggregated contrastive learning to obtain the concentrated inlier distribution as well as the dispersive outlier distribution.

Despite the promising results, the effectiveness of feature-reconstruction-based methods heavily relies on the quality of the reconstructed features, which is difficult to achieve. To more intuitively demonstrate this issue, we map the image features back into the RGB image using an image decoder[1]. The alignment between the original image and the reconstructed image determines how well the feature reconstruction captures visual differences. However, as shown in Fig. 2, we observe that UniAD (used as a representative method) fails to adequately capture crucial object details, such as edges and lighting. This limitation may lead to false positive predictions in anomaly detection and ultimately result in inferior performance.

Furthermore, we further investigate the feature reconstruction pro-

---

* Corresponding Author. Email: dinggg@tsinghua.edu.cn.

[1] The image decoder is designed to over-fit the test distribution, enabling it to perfectly map the image features back into the RGB image and reflect the quality of the reconstructed features.
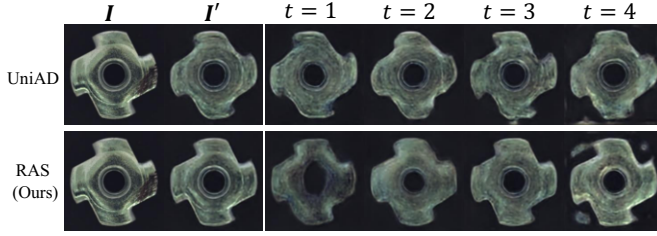
Figure 2. **Top:** inspection of the reconstruction failure of UniAD. **Bottom:** illustration of the superior reconstruction quality of our proposed RAS method. $\boldsymbol{I}$ is an anomalous metal nut.

cess step-by-step for a deeper understanding. We encourage each reconstructed feature, *i.e.,* the output of each decoder, to be mapped back into the RGB image space. By inspecting the differences between these reconstructed images, as shown in Fig. 2, we observe that there are minimal variations among the successively reconstructed images in UniAD. This indicates that the decoder fails to capture the intricate patterns already reconstructed by the preceding decoder, leading to limited contextual awareness throughout the reconstruction process. This naturally raises the question: *How to improve contextual correspondence during the feature reconstruction to enhance anomaly detection?*

In this paper, we answer this question through a novel **R**econstruction **A**s **S**equence (**RAS**) method, which rethinks the feature reconstruction process from the perspective of sequence modeling for the unified unsupervised anomaly detection. Specifically, we consider each decoder layer as one step in the sequence model. In this sense, we expect that *sequential dynamics* within different steps and *spatial dynamics* in the visual context can be simultaneously captured for the feature reconstruction. Consequently, we derive a RASFormer block that adapts the transformer architecture with a novel strategy of adaptive gating to enhance the contextual awareness ability. Benefiting from the gating strategy, our RAS method can comprehensively learn the sequential dynamics during feature reconstruction. Besides, the spatial discrepancies among the visual regions can be well grasped and enhanced for anomaly detection. As a result, our RAS can achieve superior reconstruction quality (see Fig. 2) and anomaly detection performance.

Overall, our contributions are three folds:

- We thoroughly consider the contextual awareness capability during the feature reconstruction for the unified unsupervised AD. A novel Reconstruction as Sequence (RAS) method is proposed, which rethinks the feature reconstruction process from the sequence perspective.
- We introduce a generic RASFormer block to effectively enhance the contextual correspondence during the feature reconstruction, resulting in remarkable reconstruction outcomes.
- Experimental results on several benchmark datasets show that the proposed RAS can achieve state-of-the-art performance, well demonstrating the effectiveness and superiority of the proposed method.

2 Related Work

Unsupervised anomaly detection. Due to the limited availability of anomalous samples, unsupervised learning methods are commonly employed for anomaly detection in real scenarios, *e.g.,* industrial quality inspection. Early works incorporate patch-level embedding [38], geometric transformation [17], and elastic weight consolidation [30], resulting in great improvement. Some works use a

pre-trained backbone to extract features and model the normal distribution [13, 31], followed by a distance metric to identify anomalies. Nonetheless, these methods are computationally expensive due to the need of memorizing all image features, making them impractical when facing a large number of images. Knowledge distillation methods [6, 32, 41] distinguish the difference between teacher and student for anomaly detection.

Reconstruction-based works assume that reconstruction models trained solely on normal samples perform well in normal regions but fail in anomalous regions [5, 9, 25]. Representative works include using generative networks [11, 1, 25], pseudo-anomaly [29, 12], and synthesizing anomalies on normal images [40, 24]. While these methods have shown success in separate one-class-one-model anomaly detection (AD) scenarios, their performance tends to be subpar in the unified n-class-one-model scenario [39, 42].

Unified unsupervised AD. Conventional AD methods require training separate models for each class, which becomes costly as the number of classes increases. Recently, constructing a unified model for multi-class anomaly detection has gained popularity in the research community. RegAD [19] addresses few-shot anomaly detection by training a single generalizable model, utilizing a limited number of normal images for each category during training. UniAD [39] employs a feature-reconstruction approach to pinpoint anomalous regions with the transformer architecture. OmniAL [42] presents a panel-guided method to synthesize anomalies and achieve image reconstruction using dilated channel and spatial attention blocks. These works primarily concentrate on capturing discriminative patterns that can identify anomalies by misaligning them with the normal distribution. Our RAS essentially shares a similar objective but takes it a step further by emphasizing context enhancement from a novel sequence modeling perspective. We show that our RAS can obtain remarkable reconstruction quality and thus achieving superior performance for the unified unsupervised anomaly detection.

3 Reconstruction as Sequence (RAS)

3.1 Preliminary

Image feature extraction. In the feature-reconstruction-based model, the goal is to align the reconstructed feature $\boldsymbol{f}_{\text{rec}}$ with the original image feature $\boldsymbol{f}_{\text{org}}$. To accomplish this, we employ a pre-trained convolutional neural network as the backbone for extracting the original image feature. This backbone is denoted as ϕ, and the process of deriving features from the image $\boldsymbol{I}$ can be represented as $\phi(\boldsymbol{I}) = \{\boldsymbol{f}_1, ..., \boldsymbol{f}_n\}$, where n is the number of feature levels. Consequently, for each feature level, we apply a 3×3 average pooling operation, resize them to the same size, and concatenate all the features along the channel dimension, yielding a comprehensive feature map:

$$\boldsymbol{f}_{\text{org}} \in \mathbb{R}^{C_{\text{org}} \times (H \times W)} = \text{concat}\{\boldsymbol{f}_k | k = 1, .., n\} \qquad (1)$$

where C_{org}, H, and W are the feature dimension, height and width of the feature map, respectively.

Transformer layer. Transformer [33, 14] has emerged as a foundational architecture in the field of artificial intelligence. A transformer layer comprises two essential sub-layers: the multi-head self-attention (MHSA) and the feed-forward network (FFN). To enhance training efficiency and performance, residual connections [18] and layer normalization (LN) [2] are applied to each sub-layer independently. Here, we utilize a post-LN transformer architecture [37] to construct the transformer layer:
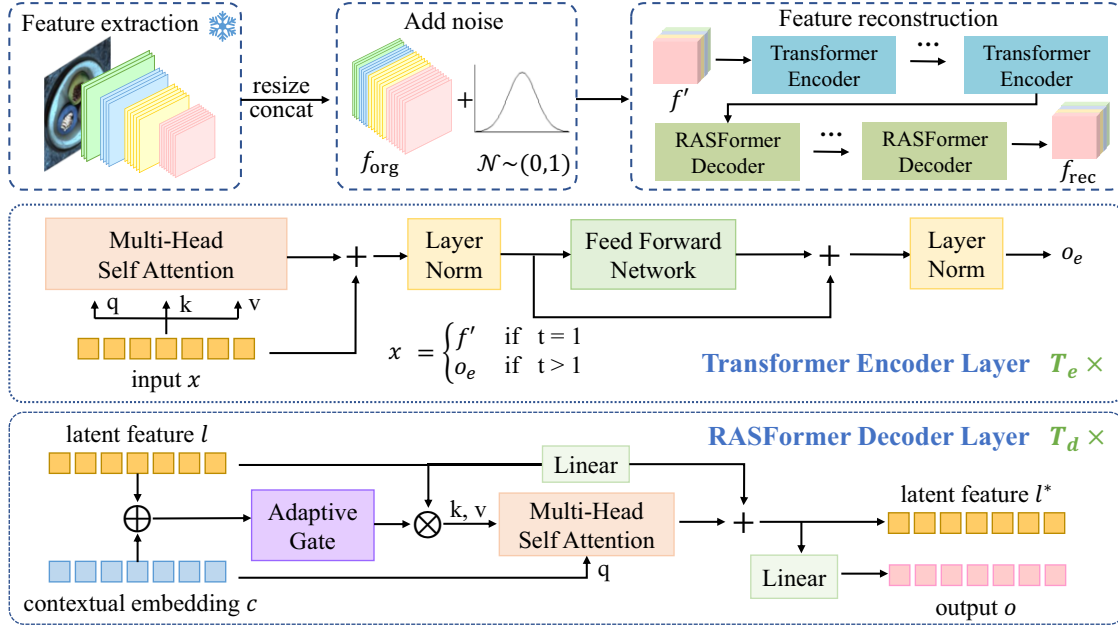
Figure 3. Overview of the proposed RAS framework for the unified unsupervised anomaly detection. We enhance the contextual awareness capability during feature reconstruction via a specially designed RASFormer block. The uppermost panel depicts the pipeline, while the two boxes below illustrate the detailed architecture of encoder-decoder to perform feature reconstruction.

$$\text{Transformer}(\boldsymbol{x}_q, \boldsymbol{x}_k, \boldsymbol{x}_v) = \text{LN}(\text{FFN}(\text{LN}(\text{MHSA}(\boldsymbol{W}_q\boldsymbol{x}_q, \boldsymbol{W}_k\boldsymbol{x}_k, \boldsymbol{W}_v\boldsymbol{x}_v)))). \tag{2}$$

where $\boldsymbol{x}_q$, $\boldsymbol{x}_k$, and $\boldsymbol{x}_v$ are input token sequences. $\boldsymbol{W}_q$, $\boldsymbol{W}_k$, and $\boldsymbol{W}_v$ are all learnable parameters. For ease of description, we omit the residual connection in the above equation.

3.2 Feature Reconstruction from the Sequence Perspective

Denoised encoding. The proposed RAS framework employs an encoder-decoder structure to reconstruct the image feature, *i.e.*, $\boldsymbol{f}_{\text{org}}$, which is derived by a CNN backbone, as depicted in Eq. 1.

We add noise to normal features and feed them into the a transformer-based encoder, achieving a robust AD model to distinguish anomalies:

$$\boldsymbol{f}' \in \mathbb{R}^{C_{\text{rec}} \times (H \times W)} = \boldsymbol{W}_f(\boldsymbol{f}_{\text{org}} + \boldsymbol{\epsilon}) \tag{3}$$

$$\boldsymbol{o}_e \in \mathbb{R}^{C_{\text{rec}} \times (H \times W)} = \text{Transformer}_{T_e}(...\text{Transformer}_1(\boldsymbol{f}', \boldsymbol{f}', \boldsymbol{f}')). \tag{4}$$

where $\boldsymbol{W}_f \in \mathbb{R}^{C_{\text{rec}} \times C_{\text{org}}}$ and C_{rec} is the dimension of the latent reconstruction space. T_e is the number of encoders. $\boldsymbol{\epsilon} = \{\boldsymbol{\epsilon}^i, i \in [0, H \times W]\}$ are the noisy features added to $\boldsymbol{f}_{\text{org}}$ during training, allowing the model to learn features of normal images through denoising:

$$\boldsymbol{\epsilon}^i \sim N(\mu = 0, \sigma^2 = (\alpha \frac{||\boldsymbol{f}_{\text{org}}^i||_2}{C_{\text{org}}})^2) \tag{5}$$

where $\boldsymbol{f}_{\text{org}}^i \in \mathbb{R}^{C_{\text{org}}}$ is one element in $\boldsymbol{f}_{\text{org}}$. α is the noise intensity to control the degree of noise. During the test phase, $\boldsymbol{\epsilon}$ is not applied.

Sequence decoding. UniAD [39] adopts conventional transformer layers to construct the decoder for feature reconstruction. Nonetheless, it is constrained in effectively capturing the contextual correspondence among decoding layers (see Fig. 2). In contrast, our proposed RAS framework can enhance the contextual correspondence by considering the feature reconstruction process from a sequence

perspective. Specifically, at each decoding step of t, we are given the previously latent features $\boldsymbol{l}_{t-1}$. Next, we equip an individual context embedding $\boldsymbol{c}_t \in \mathbb{R}^{C_{\text{rec}} \times (H \times W)}$ for each decoding step. A decoder θ_{dec}^t consumes $\boldsymbol{l}_{t-1}$ and $\boldsymbol{c}_t$ and performs the mapping from the latent space to the image feature space:

$$\boldsymbol{l}_t \in \mathbb{R}^{C_{\text{rec}} \times (H \times W)}, \boldsymbol{o}_t \in \mathbb{R}^{C_{\text{org}} \times (H \times W)} = \theta_{\text{dec}}^t(\boldsymbol{c}_t, \boldsymbol{l}_{t-1}) \tag{6}$$

where $\boldsymbol{l}_t$ is the updated latent feature and $\boldsymbol{o}_t$ is the reconstructed feature. The decoding process in Eq. 6 can be repeated several times, resulting in a sequence of reconstructions. We initialize the first latent feature with the output of the last encoder layer in Eq. 4, *i.e.*, $\boldsymbol{l}_0 = \boldsymbol{o}_e$. The final reconstructed feature $\boldsymbol{f}_{\text{rec}}$ can be denoted as $\boldsymbol{f}_{\text{rec}} = \boldsymbol{o}_{T_d}$, where T_d is the number of decoders.

It is worth noting that like the object query in DETR [8], after being well trained, the contextual embedding in the decoding step of t, *i.e.*, $\boldsymbol{c}_t$, can be considered as token query. This query provides a contextual prior assumption about t-th reconstructed features, *i.e.*, $\boldsymbol{o}_t$. During the sequence of feature reconstruction, the latent features $\boldsymbol{l}_t$ and $\boldsymbol{l}_{t-1}$ are responsible for memorization of reconstruction knowledge. Therefore, the decoder θ_{dec}^t should be powerful enough in the capability of context awareness among sequence, so that different knowledge can be uniformly captured in different decoding steps for better feature reconstruction. In light of this, we design a novel RASFormer block as the fundamental building block for decoders θ_{dec}^t. For ease of understanding, here we briefly represent the RASFormer block as a function, *i.e.*, $\theta_{\text{dec}}^t = \text{RASFormer}_t(\cdot)$.

3.3 RASFormer Block

The RASFormer block serves as a fundamental module in the decoder, playing a crucial role in capturing contextual correspondence within the sequential feature reconstruction process. We adhere to two guiding principles when designing the RASFormer block: 1) *sequential dynamics*, ensuring the awareness of the previously captured

information, alleviating the need to readdress it in subsequent reconstruction processes; 2) *spatial dynamics*, enabling the association between elements in the input context embedding c_t and those in the previous knowledge l_{t-1}. To achieve this, we introduce a novel strategy of adaptive gating with transformers.

Specifically, given the prior knowledge, *i.e.*, the previous latent reconstructed feature l and the current input context embedding c (for ease of description, we leave out the subscript t), we design an adaptive gate A to filter the prior knowledge as follows:

$$a = A(l, c) = \sigma(W_A(l \oplus c)) \tag{7}$$

$$l_A = a \otimes l \tag{8}$$

where $\oplus$ is the concatenation of two tensors along the channel dimension. $\otimes$ represents the element-wise multiplication between two matrices. W_A is a learned weight matrix. σ is an activation function (*e.g.*, sigmoid). Note that all output tensors have the same shape as l, *i.e.*, $\mathbb{R}^{C_{rec} \times (H \times W)}$. With the adaptive gate A, the current latent feature l_A can adaptively retain relevant contextual information while disregarding unimportant details. As a result, prior knowledge that is deemed irrelevant in subsequent steps can be largely disregarded, leading to the enhancement of the quality of the reconstruction output.

We then incorporate the adaptively filtered knowledge l_A with the current input information c through a transformer layer:

$$l_{Tran} = \text{Transformer}(c, l_A, l_A) \tag{9}$$

Finally, the updated latent feature l^* can be derived by fusing the previous latent feature l and l_{Tran}:

$$l^* = (Wl + l_{Tran})/2 \tag{10}$$

In order to restore it to the dimension of the original feature, we use a linear projection to get the output of the RASFormer block, which can be derived as follows:

$$o = W_o l^* \tag{11}$$

Summing it up, the RASFormer block can be summarized into a function:

$$l^*, o = \text{RASFormer}(c, l) \tag{12}$$

Remarks. The employed adaptive gate (*i.e.*, Eq. 8) can filter out the previously reconstructed information, preventing wastage of the decoder's reconstruction capacity. Also, it enables the decoder to fully consider the discrepancy between the previously reconstructed information and the currently to-be-reconstructed information, thereby achieving an enhancement of *sequential dynamics* during the reconstruction process. Furthermore, thanks to the MHSA in the transformer layer (*i.e.*, Eq. 9), the RASFormer block can facilitate the effective interaction between each element in c and other elements in l, enabling the capture of *spatial dynamics*.

3.4 Loss and Inference

Objective function. The objective function for training RAS is to calculate the MSE loss between the original feature f_{org} and the reconstructed feature f_{rec}.

$$\mathcal{L} = \frac{1}{H \times W} \|f_{org} - f_{rec}\|_2^2 \tag{13}$$

Inference. During the inference phase, the feature-level anomaly map S_{feat} is computed by measuring the L2 norm of the difference between f_{org} and f_{rec}.

$$S_{feat} = \|f_{org} - f_{rec}\|_2 \in \mathbb{R}^{H \times W} \tag{14}$$

The anomaly map is then up-sampled to the size of the original image using bi-linear interpolation to obtain the pixel-level anomaly map. The image-level anomaly score is derived by taking the maximum value of the averaged pooled pixel-level anomaly map.

4 Experiments

4.1 Experiment Setups

Datasets. We validate the effectiveness of the proposed RAS method by comparing it with several baseline methods on four widely used benchmark datasets for unsupervised anomaly detection, including MVTec-AD [4], VisA [43], BTAD [27], and MPDD [21].

- MVTec-AD is a widely used benchmark for image anomaly detection, including 15 categories of industrial products and defects. It consists of 3,629 anomaly-free images for training and 1,725 images for testing. For the test set, both normal and anomalous samples are provided (467 normal images and 1258 anomalous images).
- VisA comprises 12 subsets, each corresponding to a distinct object. It contains a total of 10,821 images, with 8,659 anomaly-free images in the training set. The test set consists of 2,162 images, including 962 normal and 1,200 anomalous images.
- BTAD presents a real-world industrial anomaly dataset, consisting of a collection of 2,540 images capturing body and surface defects in three distinct industrial products. The training set contains 1,799 normal images, and the test set includes 451 normal images and 290 anomalous images.
- MPDD includes six types of metal parts and consists of 888 images in the training set. The test set comprises 176 normal images and 282 anomalous images.

Implementation details. We employ EfficientNet-B4 as the backbone. We resize the images to 224×224 before feeding them into the backbone. Feature maps are extracted from levels 1 to 4, resulting in a concatenated feature channel C_{org} of 272. These features are aligned to the dimensions of the highest-level feature map, namely 14×14. In both the encoder and decoder, the channel dimension for the reconstructed latent feature C_{rec} is set to 256. The Multi-Head Self-Attention (MHSA) uses 8 heads. We utilize the AdamW optimizer with a learning rate of $7e - 4$ and a weight decay of $1e - 4$. The batch size is set to 64. All models are trained with 500 epochs.

Evaluation Metrics. The performance of anomaly detection models is typically measured by AUROC. We report the image-level AUROC and the pixel-level AUROC on these four datasets, following previous work [39, 6, 40].

4.2 Comparison with State-of-the-art Methods

Performance comparison on MVTec-AD. We select US [6], PaDiM [13], MKD [32], DRAEM [40], SimpleNet [26], DeST-Seg [41], UniAD [39] as our baseline methods, representing various types of anomaly detection[2]. We compare our method with baselines

---

[2] Some of the latest methods, such as PNI [3] and OmniAL [42], have heavy time and space complexities in the unified unsupervised anomaly detection setting, requiring high computation and storage. We leave them out for fair comparison.

Table 1.　Image-level AUROC for anomaly detection on MVTec-AD (unified / separate).

Category	US	PaDiM	CutPaste	MKD	DRAEM	SimpleNet	DeSTSeg	UniAD	RAS (ours)
Bottle	84.0 / 99.0	97.9 / 99.9	67.9 / 98.2	98.7 / 99.4	97.5 / 99.2	98.7 / 100	**100** / 100	99.7/ 100	**100** ± 0.00 / 100
Cable	60.0 / 86.2	70.9 / 92.7	69.2 / 81.2	78.2 / 89.2	57.8 / 91.8	93.6 / 99.9	94.5 / 97.8	95.2/ 97.6	**99.2** ± 0.12 / 99.7
Capsule	57.6 / 86.1	73.4 / 91.3	63.0 / 98.2	68.3 / 80.5	65.3 / 98.5	73.7 / 97.7	87.4 / 97.0	86.9/ 85.3	**92.6** ± 0.32 / 95.6
Carpet	86.6 / 91.6	93.8 / 99.8	93.6 / 93.9	69.8 / 79.3	98.0 / 97.0	91.5 / 99.7	98.1 / 98.9	**99.8** / 99.9	99.5 ± 0.05 / 100
Grid	69.2 / 81.0	73.9 / 96.7	93.2 / 100	83.8 / 78.0	99.3 / 99.9	50.2 / 99.7	98.4 / 99.7	98.2/ 98.5	**99.8** ± 0.16 / 100
Hazelnut	95.8 / 93.1	85.5 / 92.0	80.9 / 98.3	97.1 / 98.4	93.7 / 100	98.1 / 100	99.8 / 99.9	99.8 / 99.9	**100** ± 0.00 / 100
Leather	97.2 / 88.2	99.9 / 100	93.4 / 100	93.6 / 95.1	98.7 / 100	98.5 /100	**100** / 100	**100** / 100	**100** ± 0.00 / 100
Metal Nut	62.7 / 82.0	88.0 / 98.7	60.0 / 99.9	64.9 / 73.6	72.8 / 98.7	95.4 / 100	**100** / 99.5	99.2 / 99.0	99.9 ± 0.02 / 99.4
Pill	56.1 / 87.9	68.8 / 93.3	71.4 / 94.9	79.7 / 82.7	82.2 / 98.9	87.9 / 99.0	92.1 / 97.2	93.7 / 88.3	**96.3** ± 0.35 / 96.2
Screw	66.9 / 54.9	56.9 / 85.8	85.2 / 88.7	75.6 / 83.3	92.0 / 93.9	65.1 / 98.2	73.4 / 93.6	87.5/ 91.9	**95.3** ± 0.40 / 95.6
Tile	93.7 / 99.1	93.3 / 98.1	88.6 / 94.6	89.5 / 91.6	99.8 / 99.6	94.4 / 99.8	99.3 / 100	99.3/ 99.0	**100** ± 0.02 / 99.9
Toothbrush	57.8 / 95.3	95.3 / 96.1	63.9 / 99.4	75.3 / 92.2	90.6 / 100	85.3 / 99.7	81.7 / 99.9	94.2/ 95.0	**98.7** ± 0.30 / 94.8
Transistor	61.0 / 81.8	86.6 / 97.4	57.9 / 96.1	73.4 / 85.6	74.8 / 93.1	75.9 / 100	95.0 / 98.5	**99.8**/ 100	99.2 ± 0.00 / 100
Wood	90.6 / 97.7	98.4 / 99.2	80.4 / 99.1	93.4 / 94.3	**99.8** / 99.1	97.7 / 100	100 / 97.1	98.6/ 97.9	98.7 ± 0.23 / 98.5
Zipper	78.6 / 91.9	79.7 / 90.3	93.5 / 99.9	87.4 / 93.2	**98.8** / 100	97.8 / 99.9	99.0 / 100	95.8 / 96.7	98.4 ± 0.07 99.4
Mean	74.5 / 87.7	84.2 / 95.5	77.5 / 96.1	81.9 / 87.8	88.1 / 98.0	86.9 / 99.6	94.6 / 98.6	96.5 / 96.6	**98.4** ± 0.08 / 98.6

Table 2.　Pixel-level AUROC for anomaly localization on MVTec-AD (unified / separate).

Category	US	PaDiM	FCDD	MKD	DRAEM	SimpleNet	DeSTSeg	UniAD	RAS (ours)
Bottle	67.9 / 97.8	96.1 / 98.2	56.0 / 97	91.8 / 96.3	87.6 / 99.1	96.5 / 98.0	98.2 / 99.2	98.1 / 98.1	**98.4** ± 0.02 / 98.5
Cable	78.3 / 91.9	81.0 / 96.7	64.1 / 90	89.3 / 82.4	71.3 / 94.7	91.1 / 97.6	93.5 / 97.3	97.3 / 96.8	**98.7** ± 0.03 / 98.6
Capsule	85.5 / 96.8	96.9 / 98.6	67.6 / 93	88.3 / 95.9	50.5 / 94.3	92.2 / 98.9	96.9 / 99.1	98.5 / 97.9	**98.6** ± 0.01 / 98.6
Carpet	88.7 / 93.5	97.6 / 99.0	68.6 / 96	95.5 / 95.6	**98.6** / 95.5	96.0 / 98.2	97.4 / 96.1	98.5/ 98.0	97.9 ± 0.07 / 98.7
Grid	64.5 / 89.9	71.0 / 97.1	65.8 / 91	82.3 / 91.8	**98.7** / 99.7	53.7 / 98.8	96.6 / 99.1	96.5 / 94.6	97.1 ± 0.03 / 97.2
Hazelnut	93.7 / 98.2	96.3 / 98.1	79.3 / 95	91.2 / 94.6	96.9 / 99.7	94.8 / 97.9	**99.0** / 99.6	98.1 / 98.8	98.5 ± 0.02 / 98.7
Leather	95.4 / 97.8	84.8 / 99.0	66.3 / 98	96.7 / 98.1	97.3 / 98.6	97.1 / 99.2	**99.6** / 99.7	98.8 / 98.3	98.7 ± 0.05 / 99.2
Metal Nut	76.6 / 97.2	84.8 / 97.3	57.5 / 94	64.2 / 86.4	62.2 / 99.5	94.3 / 98.8	97.0 / 98.6	94.8 / 95.7	**97.3** ± 0.12 / 98.1
Pill	80.3 / 96.5	87.7 / 95.7	65.9 / 81	69.7 / 89.6	94.4 / 97.6	92.5 / 98.6	97.4 / 98.7	95.0 / 95.1	**98.3** ± 0.11 / 98.2
Screw	90.8 / 97.4	94.1 / 98.4	67.2 / 86	92.1 / 96.0	95.5 / 97.6	94.5 / 99.3	94.6 / 98.5	98.3 / 97.4	**99.1** ± 0.03 / 99.1
Tile	82.7 / 92.5	80.5 / 94.1	59.3 / 91	85.3 / 82.8	**98.0** / 99.2	90.9 / 97.0	95.3 / 98.0	91.8 / 91.8	92.9 ± 0.14 / 94.1
Toothbrush	86.9 / 97.9	95.6 / 98.8	60.8 / 94	88.9 / 96.1	94.7 / 98.1	94.2 / 98.5	97.7 / 99.3	**98.4** / 97.8	**98.4** ± 0.01 / 98.5
Transistor	68.3 / 73.7	92.3 / 97.6	54.2 / 88	71.7 / 76.5	64.5 / 90.9	84.5 / 97.6	78.8 / 89.1	97.9 / 98.7	**98.9** ± 0.03 / 99.1
Wood	83.3 / 92.1	89.1 / 94.1	53.3 / 88	80.5 / 84.8	96.0 / 96.4	90.7 / 94.5	**97.9** / 97.7	93.2 / 93.4	92.0 ± 0.23 / 92.9
Zipper	84.2 / 95.6	94.8 / 98.4	63.0 / 92	86.1 / 93.9	**98.3** / 98.8	96.2 / 98.9	98.0 / 99.1	96.8 / 96.0	97.8 ± 0.03 / 97.7
Mean	81.8 / 93.9	89.5 / 97.4	63.3 / 92	84.9 / 90.7	87.2 / 97.3	90.6 / 98.1	95.9 / 97.9	96.8 / 96.6	**97.5** ± 0.01 / 97.8

under the two different paradigms mentioned in Fig 1, *i.e.,* the *unified* setting and the *separate* setting. We report the performance at the image level and pixel level on MVTec-AD in Table 1 and Table 2, respectively. We can see that, for the *unified* unsupervised anomaly detection, our method can outperform UniAD with a significant improvement of 1.9% AUROC for image-level anomaly detection and of 0.7% AUROC for pixel-level anomaly localization. Although our RAS is not specifically designed for the conventional *separate* setting, it achieves comparable performance to conventional advanced methods. Compared to UniAD, our method can also obtain a 2.0% improvement in image-level AUROC and a 1.2% increase in terms of the pixel-level AUROC in the separate setting.

Performance comparison on VisA, BTAD and MPDD. For these datasets, we select DRAEM [40], SimpleNet [26], and DeST-Seg [41] as baselines due to their remarkable performance under the unified setting. Apart from results of individual dataset, we also report their average as the overall performance. Table 3 shows the comparison results. We can observe that RAS presents superior performance to baseline methods on average. Compared to DeSTSeg, our method can achieve an average improvement of 0.5% and 1.4% in image-level and pixel-level AUROC, respectively. Such performance gains across different datasets well demonstrate the effectiveness and superiority of our method.

4.3　Model Analysis

Ablation study. We analyze the impact of the adaptive gating strategy and the transformer in the RASFormer block. As shown in Table 4, with only the adaptive gate, the performance will reduce by 0.9% and 0.2% in terms of the image-level AUROC and the pixel-level AUROC, respectively. Using only the transformer can greatly decrease the performance to 96.1%/97.2%, which is 2.3%/0.3% behind our RAS (the fourth row in Table 4). These results indicate the positive effect of the proposed adaptive gating strategy and the involvement of the transformer, which can be attributed to the benefit of capturing the spatial dynamics and the sequential dynamics using them in our RAS.

Analysis of the number of encoder-decoder layers. We investigate the performance of RAS with different numbers of encoder-decoder layers in Table 5. We can observe that increasing the number of encoder layers or decoder layers can bring a substantial performance improvement. Surprisingly, feeding CNN features directly into one RASFormer decoder without the encoder, *i.e.,* $T_e = 0, T_d = 1$, can still yield quite satisfactory results (96.5%/97.2%). Adding one more decoder layer *i.e.,* $T_e = 0, T_d = 2$, can result in performance improvements, especially for the image-level AUROC (1.2%). We can also observe that compared to one decoder, more de-

Table 3.　Comparison under the unified setting on VisA, BTAD, and MPDD.

Method Dataset	DRAEM		SimpleNet		DeSTSeg		UniAD		RAS (ours)	
	I-AUC	P-AUC	I-AUC	P-AUC	I-AUC	P-AUC	I-AUC	P-AUC	I-AUC	P-AUC
VisA	74.5	84.7	87.9	95.1	88.6	96.0	88.6	98.3	**92.9**	**98.7**
BTAD	90.6	92.4	93.4	96.2	93.9	**97.2**	92.3	97.1	**94.7**	97.0
MPDD	86.9	90.6	92.5	96.3	**95.6**	95.8	87.5	95.6	92.1	**97.5**
Avg	84.0	89.2	91.3	95.9	92.7	96.3	89.5	97.0	**93.2**	**97.7**

Table 4.　Ablation study of the adaptive gate and the transformer.

adaptive gate	transformer	I-AUROC	P-AUROC
✓		97.5	97.3
	✓	96.1	97.2
✓	✓	**98.4**	**97.5**

Table 5.　Impact of the number of encoder-decoder (I-AUC / P-AUC).

	$T_d = 1$	$T_d = 2$	$T_d = 3$	$T_d = 4$
$T_e = 0$	96.5 / 97.2	97.7 / 97.3	97.9 / 97.3	98.0 / 97.4
$T_e = 1$	97.1 / 97.3	97.8 / 97.4	98.2 / 97.5	98.3 / 97.4
$T_e = 2$	97.6 / 97.4	98.0 / 97.5	98.2 / 97.5	**98.4 / 97.5**

Table 6.　Impact of noise during the feature reconstruction. Values in parentheses indicate differences from results of noise intensity of 0.

| noise intensity, *i.e.,* α		0	10	20	30	40	50
image-level	baseline	96.5	96.4 (-0.1)	96.2 (-0.3)	95.1 (-1.4)	90.9 (-5.6)	83.7 (-12.8)
	RAS	**98.4**	98.4 (**-0.0**)	98.2 (**-0.2**)	97.6 (**-0.8**)	96.6 (**-1.8**)	91.9 (**-6.5**)
pixel-level	baseline	96.8	96.8 (-0.0)	96.7 (-0.1)	96.5 (-0.3)	95.5 (-1.3)	92.0 (-4.8)
	RAS	**97.5**	97.5 (**-0.0**)	97.5 (**-0.0**)	97.4 (**-0.1**)	97.2 (**-0.3**)	96.4 (**-1.1**)

coders can lead to consistent performance improvement, indicating the remarkable benefit of modeling the feature reconstruction from the sequence perspective.

Analysis of noise during the reconstruction. Intuitively, the noise information, *i.e.,* ϵ in Eq. 5 serves as a simulation of the anomaly distribution during the feature reconstruction process. During training, our RAS can be seen as a mapping function that maps various features to the normal distribution, regardless of whether they are considered normal or abnormal. As a result, during testing (ϵ is not applied), abnormal regions can be highlighted through the difference derivation in Eq. 14. Therefore, it is worth investigating the impact of noise intensity for model inference, *i.e.,* α in Eq. 5, to gain further insights into the effectiveness of our RAS. Here, we introduce UniAD as the baseline because of its advanced performance in the unified setting on the MVTec-AD dataset. We aim to evaluate the robustness of `well-trained` models. Therefore, during testing, the noise intensity, i.e., α is adjusted to assess the noise tolerance of pretrained models. As shown in Table 6, when $\alpha = 0$, no noise is introduced for model inference, thereby achieving the best performance for both baseline and RAS. We observe that when $\alpha = 10$, RAS performs nearly on par with the condition of $\alpha = 0$, while the baseline experiences a slight 0.1% decrease in image-level AUROC. As we increase the noise scale from 20, the performance gap between the two

models gradually widens. Notably, under intense noise with $\alpha = 50$, the performance of baseline method significantly degrades, exceeding that of RAS by more than $4.0\times$ in image-level AUROC and approximately $2.0\times$ in pixel-level AUROC. Our method is more resistant to noise interference, exhibiting better robustness than baseline. This evidence clearly showcases the benefits of context enhancement by our RAS, effectively demonstrating the robustness, effectiveness, and superior ability of RAS in handling practical anomalies.

4.4　Qualitative Results

Visualization of anomaly map. To intuitively reveal the advantage of our proposed RAS model, we conduct a qualitative investigation of the anomaly maps generated by UniAD as a baseline and our RAS. As shown in Fig. 4, it is evident that our RAS can localize the anomaly regions more accurately. For instance, in examples (a), (c), and (e), our proposed method generates more accurate anomaly maps compared to the baseline. Moreover, in examples (b), (d), and (f), RAS successfully emphasizes the salience of anomalous regions by yielding higher anomaly scores. These qualitative findings effectively demonstrate the benefits of enhancing the contextual awareness capability during feature reconstruction, highlighting the superiority of our RAS.
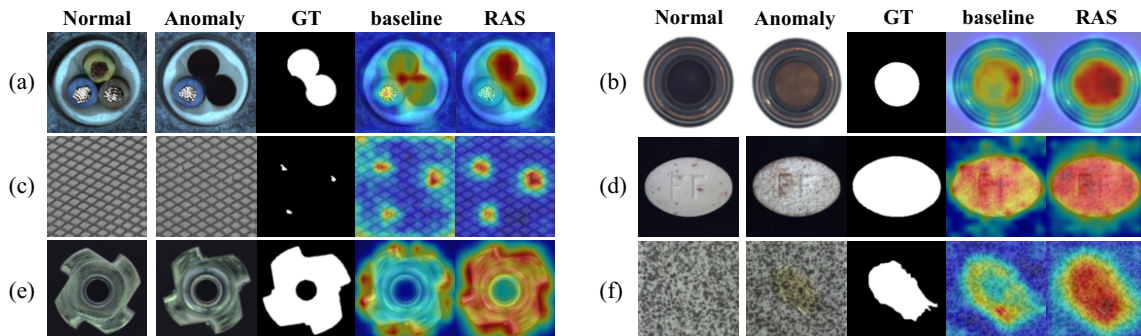
Figure 4.　**Qualitative results for anomaly map** on MVTec-AD. We turn the anomaly map into the heat map for better visualization. Regions with higher anomaly scores are depicted in vibrant red colors. Best viewed in colors. "GT" means the ground truth.
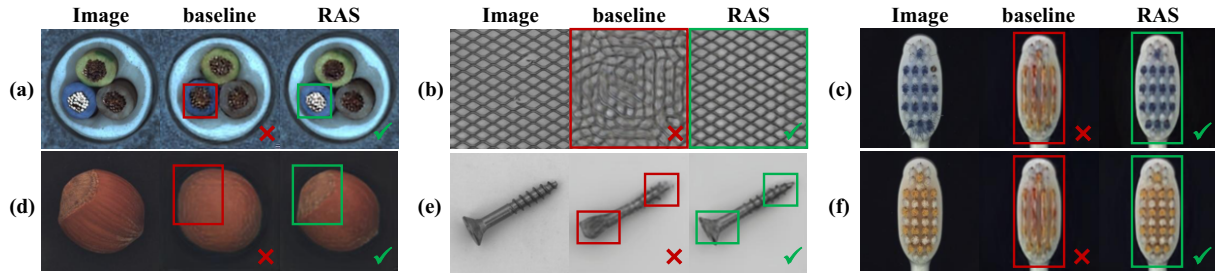
Figure 5. **Visualization comparison of image reconstruction**. We utilize bounding boxes to visually differentiate between the worse (red) and better (green) regions.
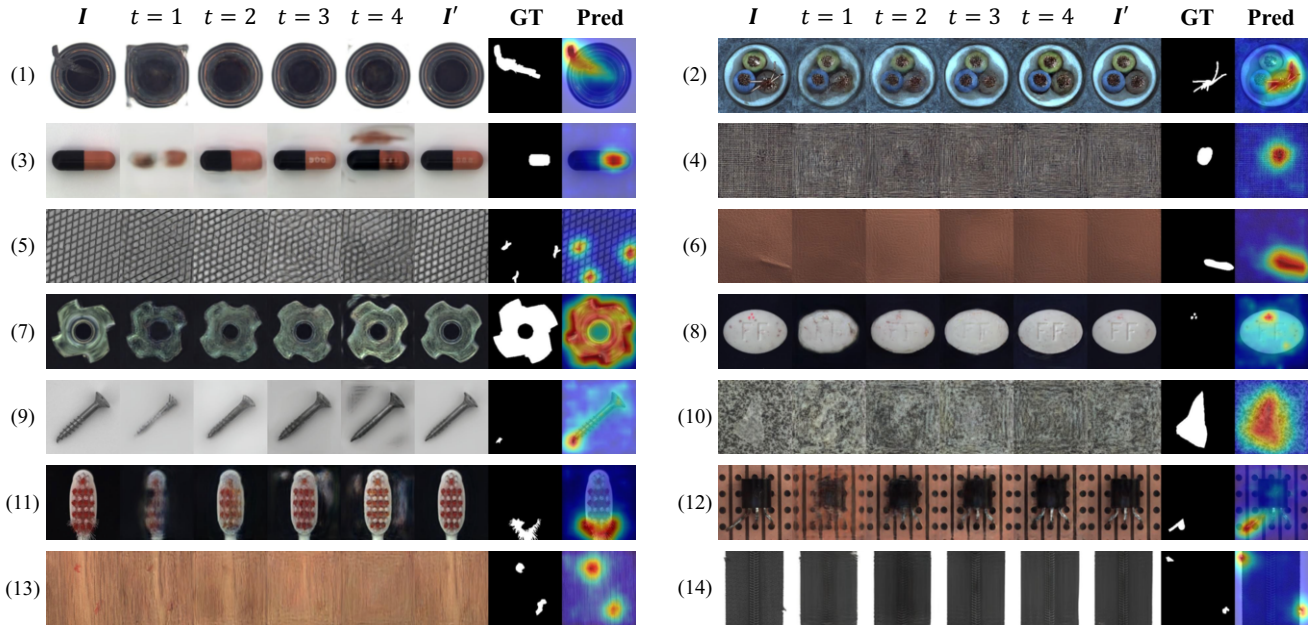

Figure 6. Visualization of the reconstruction process. I represents the original image, and t=1 to t=4 are images corresponding to the reconstructed features from each RASFormer decoder layer. I' is the final reconstructed image, "**GT**" is the binary mask, and "**Pred**" is the anomaly map predicted by our model.

Quality of image reconstruction. The superiority of our method is not only evident in the anomaly maps but also reflected in the detailed image reconstruction. Fig. 5 presents a side-by-side comparison of the reconstructed images generated by RAS and UniAD. It is clearly observed that RAS provides a more accurate reconstruction of image details. For example, in (a), RAS accurately reproduces the reflection of the cable wire in the left-bottom area. In (e), RAS correctly replicates the head and tail of the screw, while UniAD fails. These results demonstrate that the reconstructed features in RAS are more aligned with the ground truth, resulting in superior image reconstruction and anomaly detection performance.

Visualization of the reconstruction process. To better illustrate the effectiveness of the contextual awareness capability, we also visualize the images corresponding to the reconstructed features from each RASFormer decoder layer. As shown in Fig. 6, we can see that the reconstructed images effectively repair the areas where defects are present, resulting in accurate anomaly maps compared to the original images. Cases (5) and (6) demonstrate that our model can reconstruct complex textures of carpets and grids. In the case of (12), where the backgrounds are intricate, our method remains unaffected and accurately identifies anomalies in the positions of transistor pins.

By examining the features reconstructed by consecutive decoders

i.e., from $t = 1$ to $t = 4$, we can also observe that the reconstruction process in our RAS roughly follows a coarse-to-fine pattern. As a result, the output of each decoder shows a significant improvement compared to the previous time step. These results indicate that the proposed RAS can well perceive previously reconstructed information and then progressively calibrate the decoding outcome as the reconstruction process proceeds.

5 Conclusion

In this paper, we propose a novel Reconstruction as Sequence (RAS) framework for unified unsupervised anomaly detection. The main goal of our RAS is to enhance the contextual correspondence among different steps of feature reconstruction. To this end, we rethink the feature reconstruction from the sequence perspective with a generic RASFormer block. Inside the proposed RASFormer block, we adapt the transformer architecture with a novel strategy of adaptive gating. Thanks to the RASFormer block, our RAS can enhance the contextual awareness capability during feature reconstruction, leading to superior performance. Experimental results on standard benchmark datasets show that the proposed RAS can consistently outperform competing methods by a notable margin. These results well demonstrate the effectiveness and superiority of the proposed method.

Acknowledgements

References

[1] S. Akcay, A. Atapour-Abarghouei, and T. P. Breckon. GANomaly: Semi-supervised anomaly detection via adversarial training. In *Asian Conf. Comput. Vis.*, 2018.

[2] J. L. Ba, J. R. Kiros, and G. E. Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

[3] J. Bae, J.-H. Lee, and S. Kim. Pni: industrial anomaly detection using position and neighborhood information. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6373–6383, 2023.

[4] P. Bergmann, M. Fauser, D. Sattlegger, and C. Steger. MVTec AD–A comprehensive real-world dataset for unsupervised anomaly detection. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019.

[5] P. Bergmann, S. Löwe, M. Fauser, D. Sattlegger, and C. Steger. Improving unsupervised defect segmentation by applying structural similarity to autoencoders. In *International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP)*, 2019.

[6] P. Bergmann, M. Fauser, D. Sattlegger, and C. Steger. Uninformed students: Student-teacher anomaly detection with discriminative latent embeddings. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2020.

[7] E. A. L. M. Btoush, X. Zhou, R. Gururajan, K. C. Chan, R. Genrich, and P. Sankaran. A systematic review of literature on credit card cyber fraud detection using machine and deep learning. *PeerJ Computer Science*, 9:e1278, 2023.

[8] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020.

[9] L. Chen, Z. You, N. Zhang, J. Xi, and X. Le. UTRAD: Anomaly detection and localization with U-transformer. *Neural Networks*, 2022.

[10] Y. Chen, Z. Liu, B. Zhang, W. Fok, X. Qi, and Y.-C. Wu. Mgfn: Magnitude-contrastive glance-and-focus network for weakly-supervised video anomaly detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 387–395, 2023.

[11] 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 *Int. Conf. Pattern Recog.*, 2021.

[12] 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 *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 7915–7922. IEEE, 2021.

[13] T. Defard, A. Setkov, A. Loesch, and R. Audigier. PaDim: A patch distribution modeling framework for anomaly detection and localization. In *Int. Conf. Pattern Recog.*, 2021.

[14] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[15] H. Fanai and H. Abbasimehr. A novel combined approach based on deep autoencoder and deep classifiers for credit card fraud detection. *Expert Systems with Applications*, 217:119562, 2023.

[16] T. Fernando, H. Gammulle, S. Denman, S. Sridharan, and C. Fookes. Deep learning for medical anomaly detection–a survey. *ACM Computing Surveys (CSUR)*, 2021.

[17] I. Golan and R. El-Yaniv. Deep anomaly detection using geometric transformations. In *Adv. Neural Inform. Process. Syst.*, 2018.

[18] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2016.

[19] C. Huang, H. Guan, A. Jiang, Y. Zhang, M. Spratlin, and Y. Wang. Registration based few-shot anomaly detection. In *European Conference on Computer Vision (ECCV)*, 2022.

[20] Z. Huang, H. Zheng, C. Li, and C. Che. Application of machine learning-based k-means clustering for financial fraud detection. *Academic Journal of Science and Technology*, 10(1):33–39, 2024.

[21] S. Jezek, M. Jonak, R. Burget, P. Dvorak, and M. Skotak. Deep learning-based defect detection of metal parts: evaluating current methods in complex conditions. In *2021 13th International congress on ultra modern telecommunications and control systems and workshops (ICUMT)*, pages 66–71. IEEE, 2021.

[22] H. Karami, M. Kamruzzaman, J. A. Covington, M. Hassouna, Y. Darvishi, M. Ueland, S. Fuentes, and M. Gancarz. Advanced evaluation techniques: Gas sensor networks, machine learning, and chemometrics for fraud detection in plant and animal products. *Sensors and Actuators A: Physical*, page 115192, 2024.

[23] K. H. Le, T. V. Tran, H. H. Pham, H. T. Nguyen, T. T. Le, and H. Q. Nguyen. Learning from multiple expert annotators for enhancing anomaly detection in medical image analysis. *IEEE Access*, 11:14105–14114, 2023.

[24] C.-L. Li, K. Sohn, J. Yoon, and T. Pfister. CutPaste: Self-supervised learning for anomaly detection and localization. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2021.

[25] W. Liu, R. Li, M. Zheng, S. Karanam, Z. Wu, B. Bhanu, R. J. Radke, and O. Camps. Towards visually explaining variational autoencoders. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2020.

[26] Z. Liu, Y. Zhou, Y. Xu, and Z. Wang. Simplenet: A simple network for image anomaly detection and localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20402–20411, 2023.

[27] P. Mishra, R. Verk, D. Fornasier, C. Piciarelli, and G. L. Foresti. Vt-adl: A vision transformer network for image anomaly detection and localization. In *2021 IEEE 30th International Symposium on Industrial Electronics (ISIE)*, pages 01–06. IEEE, 2021.

[28] X. Peng, H. Wen, Y. Luo, X. Zhou, K. Yu, Y. Wang, and Z. Wu. Learning weakly supervised audio-visual violence detection in hyperbolic space. *arXiv preprint arXiv:2305.18797*, 2023.

[29] M. Pourreza, B. Mohammadi, M. Khaki, S. Bouindour, H. Snoussi, and M. Sabokrou. G2d: Generate to detect anomaly. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2003–2012, 2021.

[30] T. Reiss, N. Cohen, L. Bergman, and Y. Hoshen. Panda: Adapting pretrained features for anomaly detection and segmentation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2021.

[31] O. Rippel, P. Mertens, and D. Merhof. Modeling the distribution of normal data in pretrained deep features for anomaly detection. In *Int. Conf. Pattern Recog.*, 2021.

[32] M. Salehi, N. Sadjadi, S. Baselizadeh, M. H. Rohban, and H. R. Rabiee. Multiresolution knowledge distillation for anomaly detection. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2021.

[33] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. In *Adv. Neural Inform. Process. Syst.*, 2017.

[34] G. Wang, Y. Wang, J. Qin, D. Zhang, X. Bao, and D. Huang. Unilaterally aggregated contrastive learning with hierarchical augmentation for anomaly detection. *arXiv preprint arXiv:2308.10155*, 2023.

[35] J. Wolleb, F. Bieder, R. Sandkühler, and P. C. Cattin. Diffusion models for medical anomaly detection. In *International Conference on Medical image computing and computer-assisted intervention*, pages 35–45. Springer, 2022.

[36] J. Xiao and G. Ji. Divide and conquer in video anomaly detection: A comprehensive review and new approach. In *2023 China Automation Congress (CAC)*, pages 8553–8558. IEEE, 2023.

[37] R. Xiong, Y. Yang, D. He, K. Zheng, S. Zheng, C. Xing, H. Zhang, Y. Lan, L. Wang, and T. Liu. On layer normalization in the transformer architecture. In *International Conference on Machine Learning*, pages 10524–10533. PMLR, 2020.

[38] J. Yi and S. Yoon. Patch SVDD: Patch-level SVDD for anomaly detection and segmentation. In *Asian Conf. Comput. Vis.*, 2020.

[39] Z. You, L. Cui, Y. Shen, K. Yang, X. Lu, Y. Zheng, and X. Le. A unified model for multi-class anomaly detection. *Advances in Neural Information Processing Systems*, 35:4571–4584, 2022.

[40] V. Zavrtanik, M. Kristan, and D. Skočaj. DRAEM-A discriminatively trained reconstruction embedding for surface anomaly detection. In *Int. Conf. Comput. Vis.*, 2021.

[41] X. Zhang, S. Li, X. Li, P. Huang, J. Shan, and T. Chen. Destseg: Segmentation guided denoising student-teacher for anomaly detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3914–3923, 2023.

[42] Y. Zhao. Omnial: A unified cnn framework for unsupervised anomaly localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3924–3933, 2023.

[43] Y. Zou, J. Jeong, L. Pemula, D. Zhang, and O. Dabeer. Spot-the-difference self-supervised pre-training for anomaly detection and segmentation. In *European Conference on Computer Vision*, pages 392–408. Springer, 2022.