2025 International Conference on Intelligent Computing
July 26-29, Ningbo, China
https://www.ic-icc.cn/2025/index.php

MAPL: Memory Augmentation and Pseudo-Labeling for Semi-Supervised Anomaly Detection

Junzhuo Chen[1][0009-0003-7276-6507] and Shitong Kang[1][0009-0004-0441-8989]

[1] Hebei University of Technology, Tianjin 300401 , China
jzchen7@foxmail.com

Abstract. Large unlabeled data and difficult-to-identify anomalies are the urgent issues need to overcome in most industrial scene. A new methodology for detecting surface defects in industrial settings is introduced, referred to as Memory Augmentation and Pseudo-Labeling(MAPL). The methodology first introduces an anomaly simulation strategy, which significantly improves the model's ability to recognize rare or unknown anomaly types by generating simulated anomaly samples. A pseudo-labeler is employed to address the lack of labels for simulated anomalies, enhancing model robustness with limited labeled data. Meanwhile, a memory-enhanced learning mechanism is introduced to effectively predict abnormal regions by analyzing the difference between the input samples and the normal samples in the memory pool. MAPL employs an end-to-end learning framework to directly identify abnormal regions, optimizing detection efficiency. By conducting extensive trials on the recently developed BHAD dataset (including MVTec AD [1], Visa [2], and MDPP [3]), MAPL achieves an average image-level AUROC score of 86.2%, demonstrating a 5.1% enhancement compared to the original MemSeg [4] model. The source code is available at *https://github.com/jzc777/MAPL.*

Keywords: Anomaly Detection, Semi-Supervised Learning, Computer Vision.

1 Introduction

With the rise of industrial automation, efficient product surface defect detection is vital for ensuring quality and production efficiency. Traditional manual inspections fall short of modern standards, driving the growth of computer vision-based automated detection, enhanced by deep learning's advances in image processing and pattern recognition.

Accurately identifying minor defects is a significant challenge of industrial anomaly detection. Traditional anomaly detection techniques mainly rely on distance measurement between data, such as reconstruction error analysis of autoencoders. This method assumes that abnormal data shows significant differences from normal data during the reconstruction stage. Specific research includes the integration of structural similarity index into autoencoders [5], dual autoencoders generative adversarial networks [6], the combination of multi-

head variational autoencoders and discriminators [7], and the use of autoencoders for Nonlinear dimensionality reduction to handle more complex data distributions [8]. Generative adversarial networks (GANs) are widely used in unsupervised anomaly detection [9]. GANs are widely used in unsupervised anomaly detection[10] , with enhanced training methods improving efficiency[11]. Although deep learning and nearest-neighbor methods have improved the detection performance of natural image models [12], identifying tiny defects in industrial images is still challenging, mainly due to differences in image distribution. Subimage anomaly detection method [13] based on depth pyramid correspondence has significantly improved positioning accuracy in industrial applications. Furthermore, although embedding-based approaches [12-15] and knowledge distillation techniques [16-22] can extract valuable knowledge from sophisticated pre-trained models to improve the accuracy of anomaly detection, they face challenges in effectively transferring this knowledge to specific industrial applications. Existing semi-supervised and GAN-based approaches, despite progress, depend heavily on labeled data and clear anomaly distinctions, limiting their effectiveness with subtle defects and sparse labels.

To address these challenges, this paper proposes MAPL, a semi-supervised method integrating anomaly simulation, pseudo-labeling, and memory augmentation. The method utilizes an adapted VAE/GAN [24] encoder and memory augmentation network to directly identify anomalous regions in the input image by simulating anomalous samples, which optimizes the inference process and meets the real-time requirements. Moreover, by incorporating SPADE's [25] pseudo-labeling method, MAPL not only simplifies the training process but also employs consistency checks to optimize pseudo-labeling, effectively utilizing simulated anomalous samples to boost data efficiency and model generalization.

We evaluate MAPL on BHAD, a new dataset built from MVTec AD, Visa, and MDPP, with added noise to mimic industrial interference, increasing complexity and practical relevance.

In summary, the main contributions of this paper include the following four aspects:

- Proposed MAPL, a combination of semi-supervised learning method and pseudo-labeling method, effectively improves the model's ability to process unlabeled large-scale data and identify minor anomalies.
- The encoder structure based on VAE/GAN is adopted and the LeakyReLU activation function is uniformly used to optimize the accuracy of industrial defect detection..
- The anomaly simulation strategy is adopted to generate abnormal samples, which are used to train the anomaly detection module and pseudo-labeler, improving the generalization ability of the model..
- Experiments on the newly constructed BHAD data set verified the high accuracy and robustness of the MAPL method in industrial defect detection.

2 Methodology

This section presents the proposed new model, called MAPL, which is illustrated in Fig. 1. MAPL utilizes an encoder-decoder structure [26] that incorporates encoders and pseudo labeler method to address the absence of conventional methods that just rely on normal samples. During the training phase, MAPL improves semantic segmentation by incorporating simulated abnormal samples. Additionally, it enhances memory information using pseudo labelers, resulting in more precise localization of abnormal regions in images during the inference phase. Furthermore, MAPL employs enhanced encoder and memory modules to enhance its capacity to adapt and precisely manage previously unseen sorts of anomalies.

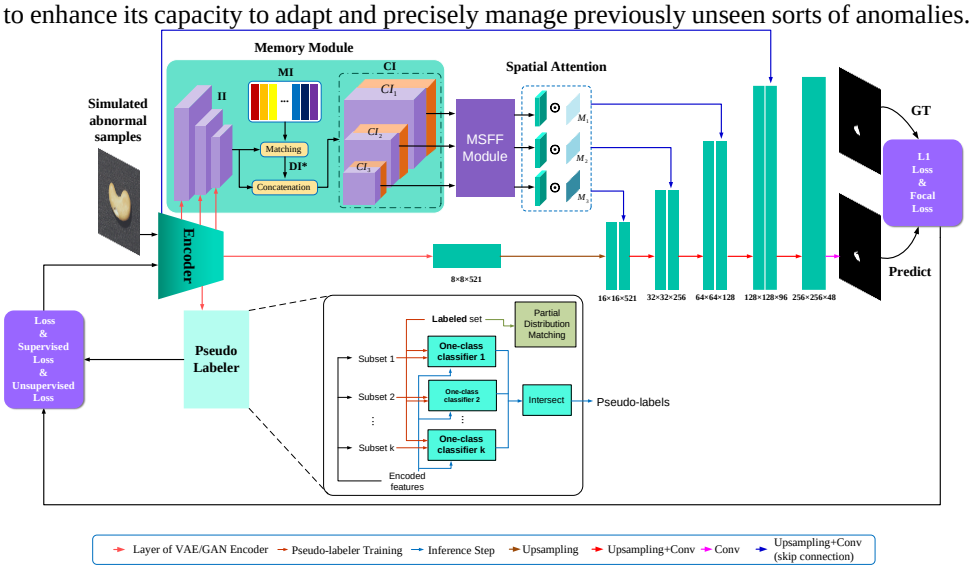

Fig. 1. Overview of the MAPL model. The model is based on the Memseg framework and incorporates an adapted VAE/GAN encoder and LeakyReLu activation function, and introduces a pseudo labeler to enhance robustness, improving the adaptability to anomaly simulation datasets and anomaly detection efficacy.

During the training phase, an anomaly simulation method is used by MAPL to incorporate artificially generated anomalous samples. These samples are created by changing the texture and structure of the normal samples I, resulting in the anomalous simulation dataset D^{a}. The model is trained using these samples to distinguish between normal and anomalous samples.

The encoder h transforms the input feature $\mathbf{x}$ into the potential representation $\mathbf{r} = h(\mathbf{x})$. For this investigation, characteristics were extracted from the input image using a modified VAE/GAN-based encoder h. To enhance the model's ability to handle negative input values, the conventional ReLU activation function was replaced with the LeakyReLU activation function [27] in the encoder. The $\mathbf{r}$ anomaly score $q(\mathbf{r})$ is calculated by Predictor q

utilizing the acquired representations. The anomaly score $q(h(\mathbf{x}))$ is determined by both the encoder and the predictor.

Subsequently, pseudo labelers are introduced into the training process. The pseudo-labeler v, with a label set $\mathrm{H} \to \{0,1,-1\}$, assigns pseudo-labels to the anomaly-simulated data $\mathbf{x}^u$ using OCCs. In this context, $v(h(\mathbf{x}^u)) = 1/0/-1$ indicates pseudo anomalous/pseudo-normal/anomaly-simulated. With this strategy, the binary cross-entropy loss L_{Y^u} is obtained for labeled data and the binary cross-entropy loss L_{Y^l} is obtained for pseudo-labeled data, by which datautilization is improved and model efficiency is enhanced.

The memory module MB stores the memory information MI extracted from normal samples, and these features are extracted by h. During the inference stage, the model calculates the L2 distance between the feature $\mathbf{x}$ of the new input image and the feature in MI to obtain the difference information DI. It then finds the minimum difference value DI^* between $\mathbf{x}$ and its nearest memory-like sample. Finally, it obtains the connection information CI, which is composed of $\mathbf{x}$ and DI^*.

Subsequently, the CI is transferred to the Multi-scale Feature Fusion (MSFF) module, where the coordinate attention method is eliminated. This module uses a 3 × 3 convolutional layer to preserve the number of channels and modify the resolution and number of channels of feature maps with varying dimensions using up-sampling and convolutional operations. It then achieves multiscale fusion by performing element-level summation.

The spatial attention map utilizes DI^* to generate three spatial attention maps, denoted as M_n. The information CI processed through MSFF is weighted through M_n and finally passed to the decoder.

Finally, the loss function module utilizes a composite loss function method, integrating L1 loss L_{l1}, focus loss L_f, and pseudo labeler-specific binary cross-entropy losses L_{Y^u} and L_{Y^l}, which collectively optimize the performance of the model.

In order to overcome these limitations, SPADE introduces the pseudo labeler method, which utilizes consistency checking to generate pseudo labels. These labels not only streamline the model training process, but also enhance the model's ability to generalize new anomaly kinds that have not been previously seen. This combination boosts both the data use efficiency and the generalization capabilities of the model, leading to improved accuracy in anomaly identification. This is particularly beneficial in complicated industrial scenarios.

2.1 Abnormal Simulation

Memseg develops a highly effective self-supervised learning approach to model anomaly samples in different types of anomaly patterns found in industrial scene. This technique is executed by a series of three sequential steps:

Initially, a binary mask M_1 is generated by converting the input picture I into a binary representation. A mask M_P consisting of consecutive blocks is created through the process of threshold binarization utilizing two-dimensional Perlin noise [28]. Ultimately, a mask image M is acquired by doing element-wise multiplication of the two produced masks.

The second step involves the creation of a noisy foreground picture, denoted as $I_n^{'}$, by merging the mask M with the noisy image I_n. Additionally, a transparency factor δ is incorporated to increase the fidelity of the simulated anomalies to the actual scenario. The mathematical expression representing $I_n^{'}$ is:

$$I_n^{'} = \delta(M \square I_n) + (1-\delta)(M \square I) \tag{1}$$

Simultaneously, the transparency of the noisy image is increased to a higher degree, and δ is selected from the range of [0.15, 1] to intensify the difficulty of model training and boost its resilience.

The background picture is obtained by multiplying the inverse mask $\bar{M}$ with the image I element by element. This background image is then added to the noisy foreground image $I_n^{'}$ to create the final simulated anomalous image I_A.

$$I_A = \bar{M} \square I + I_n^{'} \tag{2}$$

Meanwhile, the noisy image I_n incorporates the texture features from the DTD dataset [29] and the structural features obtained from random image alteration. This effectively creates simulated anomaly samples that encompass both texture and structural variations. During the training of MemSeg, the anomaly simulation dataset D^a is formed by a variable number of anomaly simulation images I_A. The number of images in the dataset is defined by the batch in the training batch. This method not only improves the resemblance between simulated and real anomalies, particularly in generating target foregrounds, but also creates the optimal conditions for the model to train effectively, hence improving the model's ability to detect anomalies.

2.2 Pseudo Labeler

2.2.1 Pseudo Labeler Architecture and Training

The model shown in Fig. 2 illustrates the configuration of the Pseudo labeler.

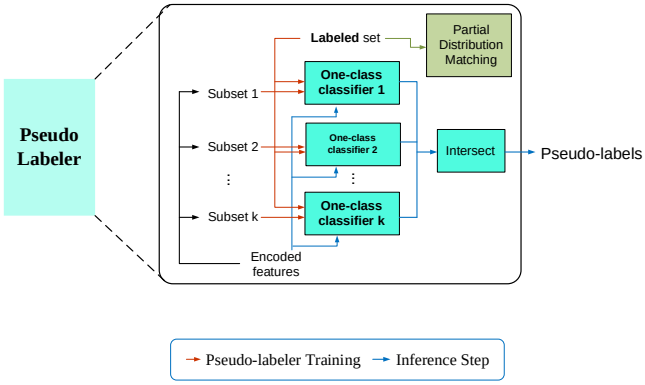

Fig. 2. Structure of Pseudo Labeler.

The pseudo-labeler comprises a model that includes K OCCs$(o_1, o_2, ..., o_K)$. Each OCC is trained with negative data (D_0^l) and one of K separate anomalous simulation subsets $(D_1^a, D_2^a, ..., D_K^a)$. The OCC generates an anomaly score, denoted as $o_k(\mathbf{x})$, for the input sample $\mathbf{x}$. When there is consensus among all OCCs, positive pseudo-labels (i.e., predicted as anomalies) are allocated to the appropriate anomalous simulation data samples: $v(h(\mathbf{x}^a)) = 1$, if $\prod_{k=1}^{K} \hat{y}_k^{pa} = 1$, where

$$\hat{y}_k^{pa} = \begin{cases} 1 & \text{if } o_k(h(\mathbf{x}^a)) > \eta_k^p \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

Likewise, the designation of negative pseudo label (i.e., normal) is assigned to a sample if all OCCs unanimously classify it as a negative pseudo label. This is based on the condition that if $\prod_{k=1}^{K} \hat{y}_k^{na} = 1$, then $v(h(\mathbf{x}^a)) = 0$, where

$$\hat{y}_k^{na} = \begin{cases} 1 & \text{if } o_k(h(\mathbf{x}^a)) < \eta_k^p \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

Simulation data that deviates from the norm and lacks agreement will be classified as unknown: $v(h(\mathbf{x}^u)) = -1$ if $\prod_{k=1}^{K} \hat{y}_k^{pa} \times \hat{y}_k^{na} = 0$

.

2.2.1 Regarding the selection of thresholds η^p and η^n

SPADE [30] utilizes the partial matching method to estimate the marginal distribution of anomalous simulated data. This method matches the distribution to a known single class (positive or negative) distribution without compromising the validation set including labeled data. The general principle states that normal samples are close to each other in feature space, while abnormal samples likewise tend to be close to each other in features. This study aims to identify the marginal distribution of positively labeled samples by comparing their distribution of abnormal scores to the distribution of abnormal scores of artificially generated abnormal samples. The value of η^p is then determined based on this comparison. The identical approach is employed to ascertain η^n by utilizing negatively labeled data. η^p and η^n are defined according to equations (5) and (6).

$$\eta_k^p = \arg\min_{\eta} D_w(\{o_k(h(\mathbf{x}^l)) \mid y^l = 1\}, \{o_k(h(\mathbf{x}^a)) > \eta\}) \tag{5}$$

$$\eta_k^n = \arg\min_{\eta} D_w(\{o_k(h(\mathbf{x}^l)) \mid y^l = 0\}, \{o_k(h(\mathbf{x}^a)) < \eta\}) \tag{6}$$

where D_w is the Wasserstein distance between the two distributions on which the determination of the subset of anomalous simulated data for pseudo-labeling is based.

2.3 Memory Module

The MemSeg model incorporates a memory module that compares observed images with stored normal samples to detect abnormal regions. A collection of selected normal images serves as memory samples, with high-level characteristics extracted by the encoder h.

Our h employs a multilayer convolutional network structure to capture image features ranging from coarse to fine. Specifically, features obtained from layers 2, 3, and 4 are utilized, which provide feature maps with dimensions N × 64 × 64 × 64, N × 128 × 32 × 32, and N × 256 × 16 × 16, respectively. Collectively, these distinct resolutions of features form what is known as memory information (MI), which supplies a wealth of visual data for detecting anomalies in the future.

h extracts dimensions of 64 × 64 × 64, 128 × 32 × 32, and 256 × 16 × 16 from the input image during the training or inference phase. The information ($\mathbf{x}$) of the input image is comprised of these multi-scale features. To calculate the L2 distance between the input image and memory sample, N difference information DI is determined using the following:

$$DI = \bigcup_{i=1}^{N} \| MI_i - \mathbf{x} \|_2 \tag{7}$$

Where N is the number of memory samples. DI is used to choose the most similar memory feature for each input characteristic in order to identify discrepancies. The minimum L2 distance signifies the existence of possibly abnormal regions, with greater values indicating a greater probability of anomalies. Additionally, it undergoes further filtration to

determine the optimal difference information DI^*, a variable that represents the minimum difference value between $\mathbf{x}$ and its closest memory sample:

$$DI^* = \underset{DI_i \in DI}{argmin} \sum_{x \in DI_i} x \tag{8}$$

The DI^* metric quantifies the extent to which the input image differs from the most similar memory samples. A higher value suggests a greater likelihood of an anomaly occurring at the corresponding region. The DI^* and $\mathbf{x}$ are combined in the channel dimension to create a spliced feature, resulting in the concatenated information CI_1, CI_2, and CI_3. The multi-scale feature fusion module processes this feature and then transfers it to the decoder via the U-Net jump link to identify anomalous regions. The multi-scale feature fusion(MSFF) module processes this feature and then it is transmitted to the decoder g for the identification of anomalous regions through the jump connection of U-Net.

2.4 Spatial Attention Map

The approach creates three spatial attention maps to predict anomalous regions and optimize disparity information utilization. The average value of the features of the various DI^* dimensions in the channel dimension is computed to create three feature maps of varying sizes (16 x 16, 32 x 32, and 64 x 64). Within this method, the M_3 spatial attention map is directly derived from the 16 x 16 feature map. M_2 is then obtained by up-sampling and multiplying M_3 at the element level by the 32 × 32 feature map. Ultimately, M_2 undergoes additional up-sampling and element-level operations are coupled with the 64 × 64 feature map to create M_1. The spatial attention maps M_1, M_2, and M_3 were individually weighted for information CI_1, CI_2, and CI_3 via MSFF processing, as seen in Fig. 1. The precise equation is as follows:

$$M_3 = \frac{1}{C_3} \sum_{i=1}^{C_3} DI_{3i}^* \tag{9}$$

$$M_2 = \frac{1}{C_2} \left(\sum_{i=1}^{C_2} DI_{2i}^* \right) \square\ M_3^U \tag{10}$$

$$M_1 = \frac{1}{C_1} \left(\sum_{i=1}^{C_1} DI_{1i}^* \right) \square\ M_2^U \tag{11}$$

Where C_n represents the number of DI_n^* channels, DI_{ni}^* represents the feature mapping of channel i in DI_n^*. M_n^U represents the feature map obtained after up-sampling M_n.

2.5 Multi-Scale Feature Fusion Module

The utilization of CI directly is hindered by feature redundancy, leading to an increased computational load on the model and thus impacting the pace of inference. Building upon the success of multiscale feature fusion in target identification [31, 32], the model utilizes a mix of a channel attention mechanism and a multiscale feature fusion method to enhance the visual and semantic information in CI.

The fusion module initiates the fusion process by employing a 3x3 convolutional layer to maintain a consistent number of channels. Despite the initial belief that the incorporation of Coordinate Attention (CA) [33] would strengthen the connection between different features, studies conducted in Section 3.5 demonstrated that it did not lead to any meaningful performance enhancement. Thus, in order to enhance efficiency, the decision was made to eliminate this method in order to decrease the computational load without sacrificing performance.

Afterwards, the feature maps of varying dimensions are aligned in terms of resolution and number of channels using up-sampling and convolution operations. Then, multi-scale feature fusion is achieved by summing the elements at each level.

Ultimately, the process of combining multi-scale features is accomplished through the inclusion of elements at a granular level. The combined feature maps are multiplied by the spatial attention map M_n and then fed into the final decoder to enhance the effectiveness of the model in decoding intricate situations.

2.6 Loss Function and Optimization

2.6.1 Composite Loss Function Design and Application

In this study, L1 loss and focal loss [34] are utilized in the Memseg model. L1 loss preserves image edge details, while focal loss balances weights between normal and aberrant regions, improving segmentation of challenging samples. The formula is as stated:

$$L_{l1} = \| G - \hat{G} \|_1 \tag{12}$$

$$L_f = -\alpha_t (1 - p_t)^{\gamma} log(p_t) \tag{13}$$

where G is the ground truth of the anomalous region in the anomalous simulation dataset, and $\hat{G}$ is the predicted value. p_t denotes the predicted probability p when the corresponding pixel has a true label of 1 in G, and similarly $p_t = 1 - p$ if the true label of G is 0. Meanwhile, the two hyperparameters, γ and α_t, are utilized to fine-tune the intensity of the weight application.

Ultimately, these limitations are combined to form an objective function:

$$L_m = \omega_{l1} L_{l1} + \omega_f L_f \tag{14}$$

where ω_{l1} and ω_f are the weighting factors for the different loss functions. Initially, in the training phase, a fundamental composite objective function (Eqs. 14) is established. This function is adjusted to cater to the diverse optimization requirements of the model by merging the L1 loss and the focus loss. Furthermore, in order to improve the model's capacity to handle abnormal simulated data, the loss function of the pseudo labeler is integrated into the overall objective function.

2.6.2 Pseudo Labeler Loss Function Strategy

In the training phase of pseudo-labeling, two types of loss functions are utilized: a supervised binary cross entropy (BCE) loss [35] for labeled data, and an unsupervised BCE loss for tackling pseudo-labeled data. This combination enhances the overall efficacy of the model in both supervised and unsupervised learning situations. More precisely, the supervised loss is determined by the binary cross entropy (BCE) loss function, which is used on labeled data to guarantee that the model's predictions align with the true labels. The purpose of this loss function is to incentivize the model to effectively differentiate between normal and abnormal samples.

$$L_{Y^l} = \mathrm{E}[L_{BCE}(q(h(\mathbf{x}^l)), y^l)] \tag{15}$$

where $q(h(\mathbf{x}^l)$ denotes the $\mathbf{x}^l$ prediction result of the model for labeled data and y^l is the corresponding true label.

The unsupervised loss is utilized on pseudo-labeled data, and the automatically generated labels are employed for training the model in order to enhance its ability to generalize to unfamiliar data.

$$L_{Y^u} = \mathrm{E}[L_{BCE}(q(h(\mathbf{x}^u)), v(h(\mathbf{x}^u))) \times 1\{v^u \in \{0,1\}\}] \tag{16}$$

where $v(h(\mathbf{x}^u))$ represents the pseudo-labeled prediction results for the anomalous simulated data $\mathbf{x}^u$, generated by Pseudo Labeler. $1\{v^u \in \{0,1\}\}$ is a binary weight that adaptively modifies the influence of the loss incurred from pseudo-labeled data based on the present performance of the model.

The model aims to minimize combined losses, including Memseg loss, mathematically represented as:

$$L_{all} = L_m + L_{Y^u} + L_{Y^l} \tag{17}$$

2.6.3 Training optimization and convergence monitoring

During model training, the combined loss function L_{all} is applied, designed to optimize encoder (h) , predictor (q), Memory bank (MB), MSFF (MF) and Decoder (g).

$$h^*, q^*, MB^*, MF^*, g^* = \arg \min_{h,q,MB,MF,d} L_{all} \tag{18}$$

Throughout the training phase, the fluctuation in loss is carefully observed. If the drop in loss does not exhibit a significant decline during multiple consecutive training iterations, it might be inferred that the model is nearing convergence. At this juncture, the option is available to either end the training process or modify parameters, such as the learning rate, in order to enhance the performance of the model.

3 Experiment

3.1 Dataset

To accurately assess the effectiveness and feasibility of the proposed MAPL method, this study developed a new dataset called BHAD (Benchmark for High precision Anomaly Detection)[36], publicly available on Mendeley Data, to validate the MAPL method. BHAD integrates MVTec AD, Visa, and MDPP datasets, ensuring training consistency. It includes four object types with training and test samples featuring diverse textures and sizes of authentic defects. Gaussian noise and contrast adjustments simulate real-world industrial interference, enhancing model robustness. Performance is evaluated using image-level AUC-ROC metrics for precision and reliability.

3.2 Experimental Methods

For an exhaustive demonstration of the experimental methodology, a pseudo-code for the MAPL model (Algorithm 1) is provided, which describes in detail the key steps from data preprocessing to model training. This includes the generation of pseudo-labels and the processing of anomalous simulated data, guaranteeing the reproducibility of the experiments.

Algorithm 1 Memory Augmentation and Pseudo-Labeling for Semi-Supervised Anomaly Detection (MAPL)

Input: Labeled training data D^l

Output: Trained encoder (h), predictor (q), Memory bank (MB), MSFF(MF), Decoder(g)

1: **function** PSEUDO-LABELER(D_1^l, D_0^l, D^a, h)

2: Split the dataset D^a into K non-overlapping subsets $\{\mathrm{D}_k^a\}_{k=1}^K$

3: **for** $k = 1:K$ **do**

4: Train OCC models o_k on $\mathrm{D}_k^a \cup \mathrm{D}_0^l$

5: Assign η_k^p / η_k^n by applying partial matching to D_1^l, D_0^l following Eqs. 5 and 6.

6: end for
7: Build pseudo-labeler v using Eqs. 3 and 4.
8: **Return** pseudo-labeler v.
9: end function
10: **function** Train_MAPL(h, q, MB, MF, g, D^l)
11: **for** each step **do**
12: **for** each batch from D^l **do**
13: Extract samples from D^l and generate anomaly simulation samples to form an anomaly simulation dataset D^a following Eqs. 1 and 2.
14: **Set** positively / negatively labeled data D_1^l, D_0^l
15: v=PSEUDO-LABELER(D_1^l, D_0^l, D^a, h)
16: Merge DI^* and D^l in the channel dimension to obtain concatenated information CI
17: Extract three spatial attention maps M using DI^* following Eqs. 9, 10, and 11.
18: Using g to predict using M to weight the CI processed by MF
19: Calculate L_m based on L1 and focal loss using Eqs. 15, 16, and 17.
20: end for
21: end for
22: **Return** h, q, MB, MF, g
23: end function
24: **Initialize** h, q, MB, MF, g
25: Extract features from N samples in D^l using h as memory samples to form a memory information MI and update MB
26: **while** h, q, MB, MF, g not converged **do**
27: h, q, MB, MF, g =Train_MAPL(h, q, MB, MF, g, D^l, D^t)
28: Update h, q, MB, MF, g using Eq. 18.
29: end while

3.3 Experimental Environment and Hyperparameter Settings

The experimental environment configuration is shown in Table 1:

Table 1. Detailed configuration parameters of experimental environment

Name	Parameter
Operating System	Ubuntu22.04
CPU	Xeon(R) Gold 6430 CPU
RAM	120 GB
GPU	NVIDIA RTX 4090（24 GB）
CUDA	12.1
Deep learning framework	PyTorch 2.1.0

The following strategies, as shown in Table 2, are adopted to ensure the efficient training of the MAPL model:

Table 2. Parameter configuration for model training

Hyper-parameters	Parameter
Iteration steps	500
Batch_size	8 (4 normal samples and 4 simulated abnormal samples)
Image_size	256×256
Learning rate	0.003
Focus loss	4
Loss function weighting factor	0.6
Loss function weighting factor	0.4
Number of memory samples	30
OCC settings	Gaussian Mixture Model (GMM)
Number of OCC	2

In this case, the learning rate is determined by grid search and is initially set to 0.003, followed by an initial adjustment using a warm up strategy, and finally the cosine annealing algorithm is applied for dynamic decay. Meanwhile, in order to balance the model's ability to handle different anomaly types, texture and structure anomaly simulations are set to be performed with equal probability. The number of training rounds of the pseudo labeler is dynamically adjusted according to the size of the dataset to ensure that the GMM can be effectively updated in each round of data iteration.

3.4 Comparative Experiment

On the BHAD dataset, MAPL outperforms the original MemSeg model in terms of image-level ROC-AUC scores. As shown in Table 3, MAPL achieves higher performance in all subcategories, especially in the Cashew and Leather categories.

Table 3. Image-level ROC-AUC % performance comparison between MAPL and MemSeg on the BHAD dataset

Datasets	Bracket black	Cashew	Leather	Screw	Average
Memseg	75.23	78.00	96.71	74.46	81.10
MAPL （Ours）	**81.07**	**87.17**	**98.78**	**76.67**	**86.20**

MAPL shows significant advantages in processing complex and texture-rich industrial images, which validates its effectiveness in enhancing model robustness and accuracy, especially in the presence of noise interference and complex textures.

3.5 Ablation Experiments

Ablation experiments play an imperative role in evaluating the impact of various model components on overall performance. This study specifically focuses on whether or not to remove the effects of CA, spatial attention maps, and MSFF. The following are the primary experiments conducted:

Table 4. Image-level ROC-AUC % performance comparison of MAPL's ablation experiments on the BHAD dataset

Datasets	Bracket black	Cashew	Leather	Screw	Average
No msff	75.23	78.00	96.71	74.46	82.28
Added CA	**81.07**	**87.17**	98.78	76.67	85.92
No Spatial Attention	46.04	78.89	**99.35**	71.12	73.85
MAPL （Ours）	78.01	85.62	99.21	**81.96**	**86.2**

The data demonstrates that removing MSFF has a substantial impact on the model's performance, resulting in a decrease in the average AUROC to 82.28%. This suggests that MSFF is imperative for feature fusion and minimizing information redundancy. Although CA enhances performance in some configurations, the overall MAPL model without CA applied demonstrates the highest average performance across all test settings. Eliminating the spatial attention maps led to a notable decline in performance on certain datasets, highlighting its significance in identifying abnormal locations.

Based on the results of the ablation experiments, the exclusion of CA and the retention of MSFF and spatial attention maps were decided upon due to their demonstrated significant effectiveness. This strategic configuration, by concentrating on the most influential components, has led to a simplified model framework and a reduction in computational burden, without substantially affecting performance, as evidenced by the highest observed average

AUROC scores. Through such a configuration strategy, the efficiency of computational resource utilization has been optimized, and the robustness and accuracy of the model in various anomaly detection scenarios have been ensured. Furthermore, the key to configuring the modules involves a systematic assessment of each component's contribution to the overall model performance, followed by selective application or exclusion to achieve an optimal balance between performance and resource utilization.

According to the above experiments, it can be obtained that the MAPL model performs optimally under all test conditions, with an average AUROC of 86.20%, which is 5.1% higher than MemSeg's 81.10%, and fully proves the synergistic effect between the model components.

4 Conclusion

The paper proposes the MAPL model, which obtains an average image-level AUROC score of 86.2% on the BHAD dataset. This model significantly improves the accuracy and resilience of industrial surface defect identification, particularly in challenging conditions with noise interference. This outcome not only demonstrates exceptional efficacy in handling intricate situations with noise interference but also presents novel methodologies for study and practical applications in the realm of industrial anomaly detection. In the real world, obtaining high-quality anomaly data is a challenge, and this challenge highlights the need to further optimize and validate the MAPL model for application in different real-world environments. Therefore, future work will concentrate on expanding the application scenarios and improving the model by incorporating additional practical environments for further validation and optimization of the MAPL model. These explorations will tackle the current restrictions and improve the model's capacity to generalize and be practically applied.

In summary, this research tackles the challenges of handling unlabeled large-scale data and detecting difficult-to-identify anomalies in industrial settings.

Acknowledgments. This study did not receive any funding.

References

1. Bergmann, P., Fauser, M., Sattlegger, D., Steger, C.: MVTec AD -- A Comprehensive Real-World Dataset for Unsupervised Anomaly Detection. Presented at the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2019).
2. Zou, Y., Jeong, J., Pemula, L., Zhang, D., Dabeer, O.: SPot-the-Difference Self-supervised Pre-training for Anomaly Detection and Segmentation. In: Computer Vision – ECCV 2022. pp. 392–408. Springer, Cham (2022).
3. Jezek, S., Jonak, M., Burget, R., Dvorak, P., Skotak, M.: 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). pp. 66–71 (2021).
4. Yang, M., Wu, P., Feng, H.: MemSeg: A semi-supervised method for image surface defect detection using differences and commonalities. Eng. Appl. Artif. Intell. 119, 105835 (2023).
5. Bergmann, P., Löwe, S., Fauser, M., Sattlegger, D., Steger, C.: Improving Unsupervised Defect Segmentation by Applying Structural Similarity to Autoencoders. In: Proceedings of the 14th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications. pp. 372–380 (2019).
6. Tang, T.-W., Kuo, W.-H., Lan, J.-H., Ding, C.-F., Hsu, H., Young, H.-T.: Anomaly Detection Neural Network with Dual Auto-Encoders GAN and Its Industrial Inspection Applications. Sensors. 20, 3336 (2020).
7. Nguyen, D.T., Lou, Z., Klar, M., Brox, T.: Anomaly Detection With Multiple-Hypotheses Predictions. In: Proceedings of the 36th International Conference on Machine Learning. pp. 4800–4809. PMLR (2019).
8. Sakurada, M., Yairi, T.: Anomaly Detection Using Autoencoders with Nonlinear Dimensionality Reduction. In: Proceedings of the MLSDA 2014 2nd Workshop on Machine Learning for Sensory Data Analysis. pp. 4–11. Association for Computing Machinery, New York, NY, USA (2014).
9. Schlegl, T., Seeböck, P., Waldstein, S.M., Schmidt-Erfurth, U., Langs, G.: Unsupervised Anomaly Detection with Generative Adversarial Networks to Guide Marker Discovery. In: Niethammer, M., Styner, M., Aylward, S., Zhu, H., Oguz, I., Yap, P.-T., and Shen, D. (eds.) Information Processing in Medical Imaging. pp. 146–157. Springer International Publishing, Cham (2017).
10. Akcay, S., Atapour-Abarghouei, A., Breckon, T.P.: GANomaly: Semi-supervised Anomaly Detection via Adversarial Training. In: Jawahar, C.V., Li, H., Mori, G., and Schindler, K. (eds.) Computer Vision – ACCV 2018. pp. 622–637. Springer International Publishing, Cham (2019).
11. Zenati, H., Foo, C.S., Lecouat, B., Manek, G., Chandrasekhar, V.R.: Efficient GAN-Based Anomaly Detection, http://arxiv.org/abs/1802.06222, (2019).
12. Bergman, L., Cohen, N., Hoshen, Y.: Deep Nearest Neighbor Anomaly Detection, https://arxiv.dosf.top/abs/2002.10445v1, last accessed 2024/03/20.
13. Cohen, N., Hoshen, Y.: Sub-Image Anomaly Detection with Deep Pyramid Correspondences, https://arxiv.dosf.top/abs/2005.02357v3, last accessed 2024/03/20.
14. Zheng, Y., Wang, X., Deng, R., Bao, T., Zhao, R., Wu, L.: Focus Your Distribution: Coarse-to-Fine Non-Contrastive Learning for Anomaly Detection and Localization. In: 2022 IEEE International Conference on Multimedia and Expo (ICME). pp. 1–6 (2022).
15. Roth, K., Pemula, L., Zepeda, J., Schölkopf, B., Brox, T., Gehler, P.: Towards Total Recall in Industrial Anomaly Detection. Presented at the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2022).
16. Defard, T., Setkov, A., Loesch, A., Audigier, R.: PaDiM: A Patch Distribution Modeling Framework for Anomaly Detection and Localization. In: Pattern Recognition. ICPR International Workshops and Challenges. pp. 475–489. Springer, Cham (2021).
17. Salehi, M., Sadjadi, N., Baselizadeh, S., Rohban, M.H., Rabiee, H.R.: Multiresolution Knowledge Distillation for Anomaly Detection. Presented at the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2021).
18. Wang, G., Han, S., Ding, E., Huang, D.: Student-Teacher Feature Pyramid Matching for Anomaly Detection, https://arxiv.dosf.top/abs/2103.04257v3, last accessed 2024/03/21.
19. Tien, T.D., Nguyen, A.T., Tran, N.H., Huy, T.D., Duong, S.T.M., Nguyen, C.D.T., Truong, S.Q.H.: Revisiting Reverse Distillation for Anomaly Detection. Presented at the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2023).

20. Bergmann, P., Fauser, M., Sattlegger, D., Steger, C.: Uninformed Students: Student-Teacher Anomaly Detection With Discriminative Latent Embeddings. Presented at the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2020).
21. Deng, H., Li, X.: Anomaly Detection via Reverse Distillation From One-Class Embedding. Presented at the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2022).
22. Cao, T., Zhu, J., Pang, G.: Anomaly Detection Under Distribution Shift. Presented at the Proceedings of the IEEE/CVF International Conference on Computer Vision (2023).
23. Zhang, X., Li, S., Li, X., Huang, P., Shan, J., Chen, T.: DeSTSeg: Segmentation Guided Denoising Student-Teacher for Anomaly Detection. Presented at the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2023).
24. Larsen, A.B.L., Sønderby, S.K., Larochelle, H., Winther, O.: Autoencoding beyond pixels using a learned similarity metric. In: Proceedings of The 33rd International Conference on Machine Learning. pp. 1558–1566. PMLR (2016).
25. Yoon, J., Sohn, K., Li, C.-L., Arik, S.O., Pfister, T.: SPADE: Semi-supervised Anomaly Detection under Distribution Mismatch, http://arxiv.org/abs/2212.00173, (2022).
26. Ronneberger, O., Fischer, P., Brox, T.: U-Net: Convolutional Networks for Biomedical Image Segmentation. In: Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015. pp. 234–241. Springer, Cham (2015).
27. Maas, A.L.: Rectifier Nonlinearities Improve Neural Network Acoustic Models. Presented at the (2013).
28. Perlin, K.: An image synthesizer. ACM SIGGRAPH Comput. Graph. 19, 287–296 (1985).
29. Cimpoi, M., Maji, S., Kokkinos, I., Mohamed, S., Vedaldi, A.: Describing Textures in the Wild. Presented at the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2014).
30. Christoffel, M., Niu, G., Sugiyama, M.: Class-prior Estimation for Learning from Positive and Unlabeled Data. In: Asian Conference on Machine Learning. pp. 221–236. PMLR (2016).
31. Chen, S., Cheng, Z., Zhang, L., Zheng, Y.: SnipeDet: Attention-guided pyramidal prediction kernels for generic object detection. Pattern Recognit. Lett. 152, 302–310 (2021).
32. Lin, T.-Y., Dollar, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature Pyramid Networks for Object Detection. Presented at the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2017).
33. Hou, Q., Zhou, D., Feng, J.: Coordinate Attention for Efficient Mobile Network Design. Presented at the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2021).
34. Lin, T.-Y., Goyal, P., Girshick, R., He, K., Dollar, P.: Focal Loss for Dense Object Detection. Presented at the Proceedings of the IEEE International Conference on Computer Vision (2017).
35. Ruby, U., Yendapalli, V.: Binary cross entropy with deep learning technique for Image classification. Int. J. Adv. Trends Comput. Sci. Eng. 9, (2020).
36. Chen, J.: Benchmark for High precision Anomaly Detection. 1, (2024).