
MMRR: Unsupervised Anomaly Detection through Multi-Level Masking and Restoration with Refinement

Anonymous Author(s)

Affiliation

Address

email

Abstract

1 Recent state-of-the-art anomaly detection algorithms mainly adopt generative mod-
2 els or approaches based on deep one-class classification. These approaches have
3 hyperparameters to balance the adversarial framework of the generative adversarial
4 network and to determine the decision boundary of the classifier. Both methods
5 show good performance, but their performance suffers from *hyperparameter sen-*
6 *sitivity*. A new category of anomaly detection methods has been proposed that
7 utilizes prior knowledge about abnormal data or pretrained features, but it is more
8 generic *not to use such side information*. In this study, we propose “*Multi-Level*
9 *Masking and Restoration with Refinement (MMRR)*”, an unsupervised-learning-
10 based anomaly detection method based on a generative model that overcomes
11 *hyperparameter sensitivity* and the need for side information. MMRR learns the
12 salient features of normal data distributions through *restoration from restricted*
13 *information via masking*, resulting in a better restoration of in-distribution data than
14 out-of-distribution data. To overcome *hyperparameter sensitivity*, we ensemble
15 restoration results from information restricted to *predefined multiple levels* instead
16 of finding a single optimal restriction level, and propose a novel mask generation
17 and refinement method to achieve hyperparameter robustness. Extensive exper-
18 imental evaluation on common benchmarks (*i.e.*, MNIST, FMNIST, CIFAR10,
19 MVTecAD) demonstrates the efficacy of the MMRR.

20 1 Introduction

21 Anomaly detection tackles the problem of detecting abnormal data with a distribu-
22 tion that is significantly different from normal data. It is an important task that enables machine learning algorithms
23 to cope with unexpected distribution in real-world tasks such as self-driving or medical imaging.
24 Anomaly detection problems are formulated assuming the unavailability of abnormal data during the
25 training process; therefore, anomaly detection models cannot be trained for the original purpose of
26 anomaly detection. With same context, *it is impossible to validate in advance whether a proposed*
27 *model performs anomaly detection well during the training process*. This means that even if the
28 anomaly detection ability of the model is significantly affected by the hyperparameter values, it is
29 impossible to find the optimal hyperparameter value through validation. Therefore, a method with a
30 robust anomaly detection performance is necessary that does not include hyperparameters that have a
31 significant influence on anomaly detection performance.

32 Three deep-learning-based leading strategies have been proposed to solve anomaly detection. The
33 first is using methods based on generative model which perform anomaly detection based on the
34 efficiency of the proposed generative models in restoring data. Early generative-model-based methods
35 failed in the anomaly detection task owing to the good generalization capability of the autoencoder
36 [37, 2]. Furthermore, to solve this problem, many studies [39, 36, 1, 8, 30, 31] inspired by generative

37 adversarial networks (GANs) [15] have attempted to create autoencoders that can only restore normal
38 data by limiting the generalization capability using an adversarial concept. The second leading
39 strategy is using deep one class classification methods [20, 35, 16, 21], which try to find the smallest
40 hypersphere surrounding only normal data in unsupervised manner. However, generative model-based
41 methods that try to restore *only* normal data well and deep one class classification methods that
42 try to find hypersphere surrounding *only* normal data have hyperparameters that have a significant
43 impact on anomaly detection performance. We define *hyperparameter sensitivity* problem as having
44 hyperparameters that significantly affect performance even in the nature of the anomaly detection
45 field where abnormal data is not available.

46 The third leading strategy is using side-information-based methods, which utilize prior knowledge
47 about the difference between normal data and abnormal data [17, 13, 12, 18, 3, 41, 45, 25, 47] or
48 utilize features [4, 28, 38, 5, 33] obtained from pretrained networks. Side-information based methods
49 have shown good performance on many benchmark datasets, but it is not common to know side
50 informations that can help distinguish normal data from abnormal data. In addition, these methods
51 suffer from massive performance degradation in a setting where used side information is not applied
52 well.

53 In this paper, we propose a novel method, Multi-Level Masking and Restoration with Refinement
54 (MMRR) that does not use side information, is based on a generative model, and avoids the *hyperpa-*
55 *rameter sensitivity* problem. The motivation behind our proposed method is that a network trained
56 to restore normal data from limited information about normal data will learn the salient features of
57 normal data. So that restoration from limited information succeeds for normal data and fails for
58 abnormal data, which makes it possible to perform anomaly detection in terms of restoration. To this
59 end, our method consists of the following two key components. First, masking, which is a process
60 that uses a mask to obtain restricted information by restricting the remaining information except for
61 the parts essential for restoration. Second, restoration, which is the process of restoring original data
62 by using only the restricted information obtained through masking.

63 For MMRR to perform anomaly detection, it is necessary to find the optimal masking level that
64 causes normal data to be restored successfully and restoration of abnormal data to fail: masking level
65 is the degree to which the mask limits information. However, to avoid the *hyperparameter sensitivity*
66 problem caused by the absence of abnormal data during training, we detected anomalies through
67 ensembles at multiple masking levels rather than finding a single optimal masking level. Our novel
68 mask generation method made it possible to ensemble at multiple masking levels by enabling the
69 manual control of the masking level of the mask, which eliminated the need for adversarial loss. In
70 addition, our mask generation method made the mask learnable such that the mask most helpful for
71 restoration at the corresponding masking level was generated, which led to better anomaly detection
72 performance.

73 However, our masking method compares the degree of restoration at the same masking level without
74 considering the complexity of each data. Therefore, masking and restoration alone often restores
75 simple abnormal data better compared with complex normal data, in which case anomaly detection
76 fails. To solve this problem, we propose an additional refinement process that eliminates the difference
77 in restoration caused by the difference in data complexity. Our contributions are as follows:

- 78 • **Hyperparameter robustness and Prior knowledge-free.** We resolve the hyperparameter sensi-
79 tivity problem that previous studies had overlooked with the proposed Multi-Level Masking and
80 Restoration. Also, we have empirically shown through experiments that Multi-Level Masking is
81 robust to hyperparameters. Furthermore, our method doesn't need any prior knowledge.
- 82 • **Experiments on benchmark datasets.** Unlike existing studies, MMRR does not strive to obtain
83 optimal anomaly detection by solving the hyperparameter sensitivity problem. Nevertheless,
84 we introduced Refinement Network considering the intrinsic complexity of data, and obtained
85 comparable performance to SOTA approaches.

86 2 Related Works

87 Classical methods proposed to solve anomaly detection include PCA [19], OC-SVM [40], SVDD
88 [42], iForest [26], and KDE [7]. Most of them perform anomaly detection using hand-crafted simple
89 functions. However, advancements in deep learning have made it easier to obtain richer and more

90 complex features of data, and thus many deep-learning-based anomaly detection studies have been
91 conducted. The following three strategies are widely used deep-learning-based anomaly detection
92 techniques.

93 **Generative-model-based methods.** Methods based on generative model begin with the assumption
94 that the generative model trained only with normal data will fail to restore abnormal data. However,
95 Sakurada and Yairi [37] and An and Cho [2] have demonstrated that a, simple autoencoder and
96 variational autoencoder sufficiently restore abnormal data, thereby leading to the failure of anomaly
97 detection. Therefore, various autoencoders for anomaly detection have been proposed that perform
98 certain tasks, such as denoising [36] and inpainting [48]. Also, there are studies that [8, 23] used
99 backpropagation to measure the distance from the manifold of the data. Many generative-model-based
100 methods are inspired by GAN and adversarial training. Some previous studies assumed networks that
101 learned normal distribution through adversarial training would not be able to restore abnormal data
102 or classify them as fake data [39, 1]. Some studies have highlighted that autoencoders have good
103 generalization capabilities and tried to design autoencoders that have limited restoration capability by
104 limiting the latent space through adversarial loss [30, 31], or by prototyping the latent space [14, 20].
105 However, most generative-model-based methods suffer from the *hyperparameter sensitivity* problem
106 because they have to find the optimal point that balances adversarial losses and other losses to obtain
107 the best anomaly detection performance, which is impossible because of the absence of abnormal
108 data.

109 **Deep one-class classification methods.** Since anomaly detection cannot use abnormal data for
110 training, it is difficult to design a classifier that distinguishes between normal data and abnormal data.
111 Ruff et al. [35] proposed a deep learning solution called SVDD [42] that seeks to find the smallest
112 hypersphere surrounding normal data. They used various constraints to prevent representation
113 collapse due to the absence of abnormal data during the training process. Hu et al. [21] proposed
114 a constraint called holistic regularization to prevent representation collapse. Some studies have
115 artificially generated abnormal data for training one-class classifiers. Goyal et al. [16] obtained,
116 artificial abnormal data through adversarial search, and Pourreza et al. [32] utilized data generated
117 from immature generator as abnormal data. Methods based on deep one class classification suffer
118 from the *hyperparameter sensitivity* problem as there are variables that significantly influence the
119 performance of anomaly detection, such as the radius variable in Goyal et al. [16].

120 **Side-information-based methods.** Self-supervised methods utilize prior knowledge of the dif-
121 ferences between normal and abnormal data. For example, some studies [13, 18, 3, 41] focused on
122 differences in terms of geometry. Golan and El-Yaniv [13] assumed that a network can learn the
123 geometric features of normal data through a learning process that predicts the geometric transformations
124 applied to normal data. They expected that a trained transform classifier will fail to predict
125 abnormal data with different geometric characteristics compared with normal data. Based on this
126 study, a method to restore transformed data [11] and methods that combined geometric concept with
127 constructive learning [6] were proposed [3, 41]. Other self-supervised methods augment normal
128 data to create synthetic abnormal data and use them to train networks that can detect locally defect
129 areas [45, 25, 47]. However, as mentioned in Goyal et al. [16], these methods rely heavily on prior
130 knowledge. Some studies have attempted to perform anomaly detection using features obtained from
131 pre-trained networks using external data. [4, 28, 38, 5, 33, 9]

132 3 Multi-Level Masking and Restoration with Refinement

133 The overall framework of the proposed method is shown in Fig. 3. In this section, we provide a
134 detailed description of our method called Multi-Level Masking and Restoration with Refinement
135 (MMRR). We describe the *Multi-Level Masking* and *Restoration* procedures that restrict the informa-
136 tion in a given input, and finally the *Refinement* that further improves the restored image.

137 3.1 Multi-Level Masking

138 Masking is a process that restricts embedding $e \in \mathbb{R}^d$, which is generated through embedding
139 network ($f_E : \mathbb{R}^d \rightarrow \mathbb{R}^d$) as $e = \tanh(f_E(x))$, by using mask m . The masking process is

$$\tilde{e} = e \odot m + \epsilon \odot (1 - m), \quad (1)$$

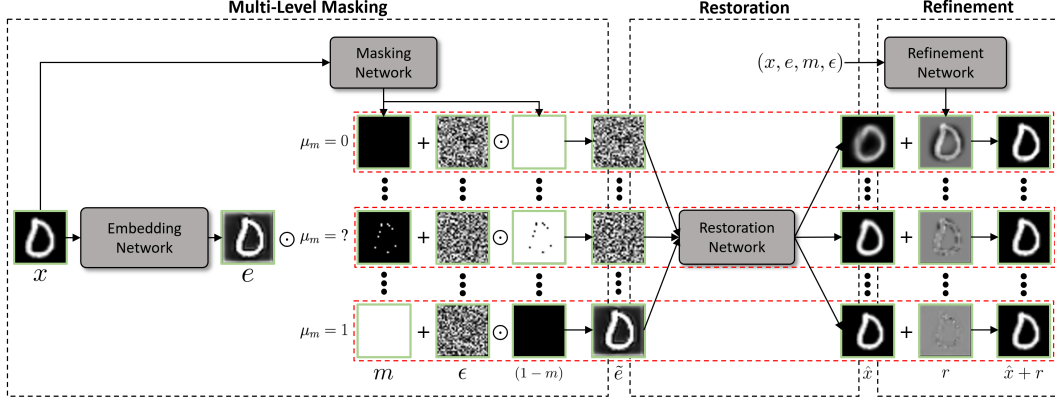


Figure 1: **Overall framework of MMRR.** Given data x , the embedding network f_E generates embedding e . The embedding thus generated is limited through a mask m with a masking level μ_m generated through the masking network f_M , and using only such restricted embedding, the restoration network f_{res} performs the restoration of the original data x . Finally, the refinement network f_{ref} complements the part not restored where restoration has inevitably failed due to the intrinsic complexity, which allows MMRR to perform anomaly detection only with the intended difference caused by masking and restoration.

140 where \odot is the Hadamard product, and ϵ is noise sampled from uniform random noise $\epsilon \sim \mathcal{U}(-1, 1)^d$.
 141 The output of the masking process, \tilde{e} , is called restricted embedding.

142 We masked e instead of directly masking x because the training process using only normal data will
 143 make f_E generate e , which helps in restoration. Thus, using e will enable our proposed masking
 144 and restoration method to have a better discrimination ability. Noise ϵ is used because, without ϵ ,
 145 irrespective how small m is, trivial solution that can easily restore data is generated because e is
 146 learnable. For the same reason, \tanh was used to create e to prevent a trivial solution that makes
 147 restoration easier by making the value of e significantly different from the noise value.

148 We can easily infer from Eq. 1 that if the value of m become smaller, the portion of embedding e in
 149 \tilde{e} decreases and becomes noisy, and restoration becomes harder. For example, if all elements of m
 150 are 0, \tilde{e} will resemble uniform noise $\mathcal{U}(-1, 1)^d$, and restoration will be impossible. Therefore, we
 151 consider that the average value of m can represent the difficulty of restoration from \tilde{e} and define it as
 152 a masking level $\mu_m = \frac{1}{d} \sum_{i=1}^d m_i$, where m_i refers to i -th element of m and $\mu_m \in [0, 1]$.

153 The restricted embedding $\tilde{e} \in \mathbb{R}^d$ should meet two conditions for masking and restoration to detect
 154 anomalies: normal data should be successfullt restore and abnormal data should not be restored. To
 155 accomplish the goal, we need to find m with a μ_m that can best differentiate normal and abnormal data
 156 in terms of restoration. However, we cannot find an optimal masking level μ_m that best distinguishes
 157 abnormal data from normal data. This is because abnormal data cannot be used in the training process
 158 owingto the nature of the field of anomaly detection.

159 Therefore, we decided to ensemble the ability to distinguish at multiple masking levels μ_m , which are
 160 uniformly distributed between 0 and 1. For example, if we use L levels of masking for the ensemble,
 161 $\mu_m \in \{0, \frac{1}{L-1}, \frac{2}{L-1}, \dots, 1\}$ will be used.

162 Our novel mask generation method made it possible to manually adjust the μ_m of m for multilevel
 163 ensemble. Furthermore, the novel mask generation made m learnable such that it is generated in a
 164 direction that is most useful for restoration from the corresponding μ_m , which improves the ability to
 165 distinguish between normal data and abnormal data.

166 **Mask generation method.** We propose a novel mask generation method that can generate a mask
 167 m with masking level μ_m by $m = \sigma(f_M(x) + b)$, where $f_M : \mathbb{R}^d \rightarrow \mathbb{R}^d$ is the masking network.
 168 The goal of our mask generation method is to find the appropriate bias $b \in \mathbb{R}$ that makes the average
 169 value of the mask to a predefined μ_m as follows: $\frac{1}{d} \sum_{i=1}^d \sigma(f_M(x)_i + b) = \mu_m$, where μ_m is on
 170 interval $[0, 1]$ because $m \in [0, 1]^d$. As sigmoid σ is a monotonically increasing function, we can use
 171 the root-finding method (in our case, the bisection method) to find bias b that satisfies the condition,

172 which allows us to successfully generate mask m with masking level μ_m . While the root finding
 173 method is non-differential, the gradient to the output of f_M was obtained under the assumption that
 174 the bias satisfying the condition was well found, which is as follows:

$$\frac{\partial \mathcal{L}}{\partial f_M(x)_i} = \sum_{\forall j} \frac{\partial \mathcal{L}}{\partial m_j} m_j (1 - m_j) \left(\delta(i, j) - \frac{m_i (1 - m_i)}{\sum_{\forall k} m_k (1 - m_k)} \right), \quad \delta(i, j) = \begin{cases} 1, & \text{if } i = j \\ 0, & \text{otherwise} \end{cases}. \quad (2)$$

175 3.2 Restoration

176 Restoration refers to the process in which restricted embedding \tilde{e} is restored to original data x via the
 177 restoration network ($f_{\text{res}}: \mathbb{R}^d \rightarrow \mathbb{R}^d$), where the restoration output is $\hat{x} = \tanh(f_{\text{res}}(\tilde{e}))$. Owing to
 178 the mask generation method, not only f_{res} but also f_E and f_M can be trained with only the simple
 179 restoration loss, which is formulated as:

$$\mathcal{L}_{\text{res}} = \frac{1}{d} \sum_{i=1}^d (x_i - \hat{x}_i)^2 \quad (3)$$

180 f_{res} , which is trained using a training dataset consisting of only normal data, learns how to restore
 181 normal data from \tilde{e} . In such a training process, f_{res} will learn salient features for normal distribution
 182 p^+ . The features for normal distribution p^+ obtained in this way will allow f_{res} to restore normal
 183 data efficiently even when the masking level μ_m is small.

184 While f_{res} has been able to successfully restore normal data as mentioned above, this way of
 185 restoration will fail for abnormal data. The reason is, f_{res} has no choice but to generate an output that
 186 resembles normal data because f_{res} will also apply learned features for p^+ even when restoration is
 187 performed from \tilde{e} of abnormal data. This failure to restore abnormal data will allow the masking and
 188 restoration method to detect anomalies through the restoration loss.

189 3.3 Refinement

190 Our masking and restoration method resolves the *hyperparameter sensitivity* problem by ensembling
 191 the anomaly detection performance at multiple masking levels $\mu_m \in \{0, \frac{1}{L-1}, \frac{2}{L-1}, \dots, 1\}$. However,
 192 comparing the degree of restoration at the same μ_m without considering the characteristics of the
 193 data causes another problem. This is because the degree of restoration is intrinsically different even if
 194 it is restored from the same μ_m because different x have different complexities.

195 Let us assume that the restoration loss obtained from masking and restoration is composed of two
 196 losses. The first loss is caused by the inevitable restoration failure due to the intrinsic complexity of
 197 x , which is denoted as intrinsic loss. The second loss occurs when abnormal data are restored like
 198 normal data owing to masking and restoration, which is denoted as abnormality loss. We originally
 199 intended to perform anomaly detection based on this abnormality loss.

200 This problem occurs when the abnormal sample is relatively simple compared with the normal sample.
 201 In this case, the sum of the intrinsic loss and the abnormality loss of the abnormal sample can be
 202 smaller than the intrinsic loss of the normal sample, which leads to the anomaly detection failure of
 203 the masking and restoration method.

204 To address this problem, the refinement method aims to eliminate the intrinsic loss that inevitably
 205 occurs due to intrinsic complexity difference so that anomaly detection can be performed only with
 206 the abnormality loss caused by masking and restoration process. For this, the refinement network
 207 $f_{\text{ref}}: \mathbb{R}^d \times \mathbb{R}^d \times \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}^d$ predicts $x - d$ that have not yet been restored at a particular μ_m as
 208 follows: $r = f_{\text{ref}}(x, e, m, \epsilon)$. f_{ref} is trained with refinement loss formulated as:

$$\mathcal{L}_{\text{ref}} = \frac{1}{d} \sum_{i=1}^d (x_i - (\hat{x}_i + r_i))^2 \quad (4)$$

209 3.4 Training and Evaluation

210 **Training.** Our method consists of a two-step training process. The first phase is a training process
 211 for masking and restoration. During this phase, the masking level μ_m is uniformly sampled, where

212 $\mu_m \sim \mathcal{U}(0, 1)$. The embedding network f_E , masking network f_M , restoration network f_{res} are trained
 213 only with restoration loss. Furthermore, we selected the model with the smallest restoration loss for
 214 the validation data. The second phase is a training process for refinement. To this end, networks that
 215 have been trained in the first phase are used with fixed weights. μ_m is sampled from $\mathcal{U}(0, 1)$ as in
 216 first phase. The refinement network f_{ref} is trained with only the refinement loss. Furthermore, we
 217 selected the model with the smallest refinement loss for the validation data.

218 **Evaluation.** For evaluation, we must first determine the number of μ_m required. If we decide to
 219 use L masking levels, we must use $\{0, \frac{1}{L-1}, \frac{2}{L-1}, \dots, 1\}$ masking levels that are distributed evenly at
 220 $1/(L-1)$ intervals for evaluation. Finally, we perform anomaly detection by summing the refinement
 221 loss at all masking levels for ensemble.

222 4 Experiment

223 4.1 Experimental Settings

224 To validate the proposed anomaly detection method, MMRR using multi-class datasets (MNIST [24],
 225 FMNIST [44], CIFAR10 [22]), which is not designated for anomaly detection, we used the one-vs-all
 226 strategy. The one-vs-all strategy selects one normal class $1 \leq c \leq C$ among C different classes.
 227 For training, we only used the training set belonging to class c . For testing, the normality score
 228 was calculated for all the data in the test set, the extent to which normal data and abnormal data are
 229 distinguished in terms of the normality score was measured using the area under receiver operating
 230 curve (AUROC). This process was repeated for all classes C to evaluate the anomaly detection model.
 231 On the other hand, in the case of the MVTecAD dataset, for each class c , the train dataset consisting
 232 of only normal data and the test dataset mixed with abnormal data are already prepared. Therefore,
 233 we trained using only the train data as a given material, and used the test dataset in the test process.

234 **Implementation details.** All proposed networks were implemented using the U-Net[34] based on
 235 the wide residual[46] blocks proposed for wide residual networks. We used group normalization for
 236 all blocks. For 32x32 datasets, we used four feature map resolutions(32x32 to 4x4). For 256x256
 237 datasets, we used five feature map resolutions(256x256 to 16x16). We used two wide residual blocks
 238 that consisted of convolutions with 128 output channels for each feature map resolution. RAdam[27]
 239 was used as the optimizer with a learning rate of 0.0001. Batch size was set as 64 and 4 for the 32x32
 240 and 256x256 datasets, respectively. The learning was decayed by a factor 0.5 if the validation loss
 241 did not decrease for 500 epochs. We split the normal training set into training and validation sets
 242 using a 95:5 ratio, and used the validation set to select the model with smallest validation loss.

243 4.2 Datasets and Results

244 **Baseline Methods.** For anomaly detection in multi-class datasets, we compared MMRR with
 245 classical approaches such as: OC-SVM [40], and KDE [29]; generative-model-based approaches such
 246 as: AnoGAN [39], OCGAN [30], $\gamma - VAE_g$ [10] and CAVGA_u [43]; deep one-class classification
 247 approaches such as: DSVDD [35], and DROCC [16]. For anomaly detection on MVTecAD dataset,
 248 we compared our MMRR with vanilla autoencoder AE, AE with skip connectins AE+skip, variational
 249 autoencoder VAE, Ganomaly[1], MemAE [14], CAVGA_u, and DAAD [20].

- 250 • **MNIST** includes a training set of 60,000 examples, and a test set of 10,000 examples. The data are
 251 28x28 handwritten digits(0-9). For simplicity they were resized to 32x32. It was used for training
 252 without any augmentations except resizing. Our MMRR model achieved averaged AUROC of
 253 0.967, which is slightly lower compared to SOTA methods. The reason our model has slightly poor
 254 performance on the MNIST dataset is that the data have a very easy distribution, so reconstruction
 255 occurs well enough even at a very low masking level μ_m . For example, in Figure 3, we can see that
 256 the digit 0 is restored well enough even if μ_m is 0.01. As such, if there is already a sample that can
 257 be restored well in the masking and restoration stage of very low μ_m , it can be seen that refinement
 258 has a limit in solving this problem.
- 259 • **FMNIST** consists of a training set of 60,000 examples, and test set of 10,000 examples, full of 10
 260 different types of fashion items. For simplicity they were resized to 32x32. It was used for training
 261 without any augmentations except resizing. MMRR greatly beats the existing SOTA performance
 262 of 0.885 AUROC of CAVGA.

- **CIFAR10** consists of 60000 32x32 color images in 10 classes, with 6000 images per class. There are 50000 training and 10000 test images. The dataset was used for training without any augmentations. As shown in Table 5, our method achieved an average AUROC performance of 0.737 on the CIFAR10 dataset, which is comparable to that of other SOTA methods: 0.742 for DROCC and 0.737 for CAVGA. Moreover, the performance obtained by our method is meaningful because it is obtained without experiencing *hyperparameter sensitivity* problem unlike other SOTA methods.
- **MVTecAD** is a dataset for benchmarking anomaly detection methods with a focus on industrial inspection. It contains over 5000 high-resolution images divided into 15 different object and texture categories. Each category comprises a set of defect-free training images and a test set of images with a variety of defects as well as images without defects. We resized all the data to 256x256. We performed two tasks on the MVTECAd dataset, image-level anomaly detection and pixel-level anomaly localization. Experimental results on MVTECAd dataset can be seen in Table 7. MMRR achieved average 0.865 AUROC for pixel-level anomaly detection and 0.844 AUROC for image-level anomaly detection, which is close to SOTA methods. We found that among the test defect-free data in the screw class, there were samples with a different distribution in terms of brightness compared to the train defect-free data. Therefore, we trained MMRR by applying brightness augmentation to the train data, and a result of 0.95 AUROC was obtained in the image-wise anomaly detection. However, we did not report the performance because we assumed that we do not know the distribution of the test data.

Hyperparameter sensitivity. As we mentioned earlier, most of the generative model based methods and deep-one class classification based methods have *hyperparameter sensitivity* problem. For example, DROCC [16] showed how sensitively the performance changes according to the radius value, which is a hyperparameter that they used to obtain negative samples. Anomaly detection performance of DROCC in CIFAR10 dataset fluctuates between 0.7-0.8 for airplane, 0.5-0.7 for deer, and 0.7-0.8 for trucks in terms of AUROC depending on the radius value. Therefore, they carefully searched for the radius value to obtain optimal anomaly detection performance. In addition to this, Akçay et al. [1] showed that the performance of their proposed model is sensitively changed according to the values of three hyperparameters that balance their losses in the CIFAR10 dataset. Also, Hou et al. [20] showed that the anomaly detection performance in MVTECAd dataset fluctuates between 0.716-0.821 based on the value of division rate (r_h & r_w) that determines the size of the query.

However, MMRR uses only one loss for each training phase. And we provide a clear criterion for model design: selecting the model with the lowest loss on validation data. Furthermore, we show how the performance of MMRR changes according to the only hyperparameter that significantly affects our performance in the Fig. 2. From Fig. 2, It can be seen that the performance of anomaly detection improves as the number of masking levels used for evaluation increases.

Prior knowledge. It has been shown in Goyal et al. [16] that the side-information based methods mentioned in Section 2 relies heavily on the prior knowledge they used. To prove this, they applied flips and small rotations of angle $\pm 30^\circ$ to CIFAR10 data during training. As can be seen in Table 1 there was a large decline in the performance (0.86 to 0.691) of the Golan and El-Yaniv [13] that used prior knowledge. On the other hand, MMRR w/o refine showed rather good performance (0.676 to 0.682), and MMRR showed 0.037 lower performance (0.737 to 0.7).

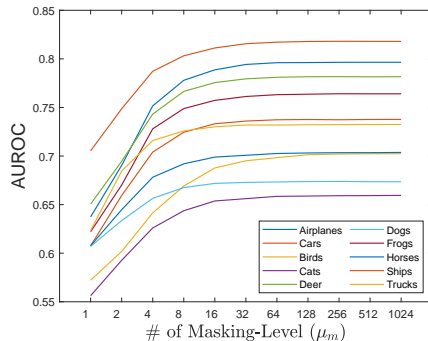


Figure 2: Illustration of AUROC with respects to the number of Masking-Level (μ_m) used for MMRR on CIFAR10 dataset.

	GEOM	MMRR w/o ref.	MMRR
w/o aug.	0.86	0.676	0.737
w/ aug.	0.691	0.682	0.7

Table 1: Comparing AUROC against GEOM[13] on CIFAR10 dataset with training data augmentations (rotation $\pm 30^\circ$ and flips).

4.3 Ablation Study

MNIST	OC-SVM	KDE	AnoGAN	DSVDD	OC-GAN	CAVGA	MMRR w/o ref.	MMRR
0	0.988	0.885	0.966	0.98	0.998	0.994	0.9857	0.9941
1	0.999	0.996	0.992	0.997	0.999	0.997	0.999	0.9982
2	0.902	0.71	0.85	0.917	0.942	0.989	0.8981	0.94
3	0.95	0.693	0.887	0.919	0.963	0.983	0.9246	0.955
4	0.955	0.844	0.894	0.949	0.975	0.977	0.9309	0.9352
5	0.968	0.776	0.883	0.885	0.98	0.968	0.9173	0.971
6	0.978	0.861	0.947	0.983	0.991	0.988	0.9765	0.989
7	0.965	0.884	0.935	0.946	0.981	0.986	0.9539	0.966
8	0.853	0.669	0.849	0.939	0.939	0.988	0.906	0.945
9	0.955	0.825	0.924	0.965	0.981	0.991	0.9511	0.98

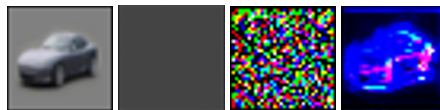
Table 5: Image-level AUROC for one-vs-all anomaly detection on MNIST.

316 **Embedding.** To prove the effectiveness of using embedding e ,
317 we directly masked the data x . As can be seen from the table, we
318 got an average AUROC of 0.6449 in the CIFAR10 dataset when e
319 was not used. And 0.6449 AUROC is far lower than 0.737 AUROC,
320 which is the performance obtained when e is used. Through these
321 results, it can be seen that f_E learned a salient features for normal
322 data in the training process of generating e , which is most helpful
323 for restoration even though it is restricted by masking. And the
324 embedding e generated from f_E can be seen to have a positive effect on the anomaly detection
325 performance by widening the restoration gap between normal data and abnormal data.

	w/o ref.	w/ ref.
w/o emb. e	0.642	0.6449
w/ emb. e	0.676	0.737

Table 2: AUROC performance of MMRR w/o and w/ embedding network on CIFAR10.

326 **Mask generation method.** We proved the effective-
327 ness of our learnable mask by comparing it with other
328 simple masks which are unable to learn. The first mask
329 is a mask in which all elements have the same constant
330 value μ_m , and we will call it a constant mask. The
331 second mask to be compared is a mask generated by
332 bernoulli sampling with a probability of μ_m . When we
333 used the constant mask, we got an AUROC performance
334 of 0.667, and when we used the bernoulli mask, we got
335 0.6478. These are lower performances when compared
336 to 0.737 obtained by our mask generation method. From
337 the experimental results, it can be seen that the use of
338 a our multi-level mask that can learn to leave informa-
339 tion which is most helpful for restoration at a specific
340 masking level during the masking process also helps anomaly detection.



(a) From left to right, data, constant mask, bernoulli mask, our mask

	Constant	Bernoulli	Ours
w/o ref.	0.619	0.612	0.676
w/ ref.	0.674	0.648	0.737

Table 3: AUROC according to mask generation method on CIFAR10

341 **Refinement.** As can be seen from
342 the Table 4, there is a big difference
343 between MMRR with refinement and
344 MMRR without refinement. In the case
345 of MNIST dataset, average auroc im-
346 proved by 0.033 from 0.944 to 0.967.
347 And for CIFAR10 dataset, average au-
348 roc improved by 0.067 from 0.68 to
349 0.747. Experimental results show another interesting phenomenon besides performance improve-
350 ment. For example, data that has already had good anomaly detection performance in MMRR w/o
351 refinement, such as data belonging to airplane, deer, ship classes, does not improve significantly when
352 refinement is applied as can be seen in Table 5. However, the data that performed poorly in the
353 MMRR w/o refinement, such as data belonging to automobile, truck, showed a remarkably large
354 performance improvement. These results show that the intrinsic complexity difference between
355 classes is well resolved through the refinement as intended. However, as MVTECAd dataset is the
356 data proposed to detect local defect areas, the difference in intrinsic complexity between normal
357 data and abnormal data is not large. Therefore, as can be seen from the Table 4, the performance
358 improvement due to refinement was insignificant.

	MNIST	FMNIST	CIFAR10	MVTecAD
w/o ref.	0.944	0.928	0.676	0.825 / 0.861
w/ ref.	0.967	0.93	0.737	0.840 / 0.865

Table 4: AUROC w/o and w/ refinement module on MNIST, FMNIST, CIFAR10, and MVTECAd. *Image-wise / Pixel-wise* AUROC performance was reported on MVTECAd.

CIFAR10	OC-SVM	KDE	AnoGAN	DSVDD	OC-GAN	γ -VAE	CAVGA	DROCC	MMRR w/o ref	MMRR
Airplane	0.63	0.658	0.671	0.617	0.757	0.702	0.653	0.817	0.7778	0.7965
Automobile	0.44	0.52	0.547	0.659	0.531	0.663	0.784	0.767	0.6065	0.7377
Bird	0.649	0.657	0.529	0.508	0.64	0.68	0.761	0.667	0.6926	0.7024
Cat	0.487	0.497	0.545	0.591	0.62	0.713	0.747	0.671	0.6076	0.6595
Deer	0.735	0.727	0.651	0.609	0.723	0.77	0.775	0.736	0.7638	0.7817
Dog	0.5	0.496	0.603	0.657	0.62	0.689	0.552	0.744	0.6143	0.6739
Frog	0.725	0.758	0.585	0.677	0.723	0.805	0.813	0.744	0.6966	0.7641
Horse	0.533	0.564	0.625	0.673	0.575	0.588	0.745	0.714	0.626	0.7037
Ship	0.649	0.68	0.758	0.759	0.82	0.813	0.801	0.800	0.7878	0.8181
Truck	0.508	0.54	0.665	0.731	0.554	0.744	0.741	0.762	0.6229	0.7325

Table 6: Image-level AUROC for one-vs-all anomaly detection on CIFAR10.

Method	Class															
	carpet	grid	leather	tile	wood	bottle	cable	capsule	hazelnut	metalnut	pill	screw	toothbrush	transistor	zipper	
Pixel-level	AE	0.539	0.96	0.751	0.476	0.63	0.909	0.732	0.786	0.976	0.88	0.885	0.979	0.971	0.906	0.68
	VAE	0.58	0.888	0.834	0.465	0.695	0.902	0.828	0.862	0.977	0.881	0.888	0.958	0.971	0.894	0.814
	γ -VAE _g	0.727	0.979	0.897	0.581	0.809	0.931	0.88	0.917	0.988	0.914	0.935	0.972	0.983	0.931	0.871
	MMRR w/o ref.	0.6733	0.8529	0.8599	0.7851	0.7911	0.8878	0.9117	0.898	0.8555	0.8648	0.9335	0.9074	0.9506	0.8865	0.8526
MMRR	0.6561	0.8477	0.8405	0.7916	0.7858	0.889	0.8841	0.9179	0.9414	0.8197	0.9209	0.8924	0.9486	0.9038	0.8779	
Image-level	Ganomaly	0.699	0.708	0.842	0.794	0.834	0.892	0.732	0.785	0.7	0.743	0.746	0.653	0.792	0.745	
	AE	0.411	0.841	0.615	0.696	0.961	0.955	0.688	0.819	0.884	0.565	0.882	0.956	0.977	0.776	0.878
	MemAE	0.454	0.946	0.611	0.63	0.967	0.954	0.694	0.831	0.891	0.537	0.883	0.992	0.972	0.793	0.871
	AE+skip	0.385	0.879	0.57	0.986	0.977	0.713	0.579	0.747	0.828	0.336	0.853	1	0.742	0.749	0.696
	DAAD	0.671	0.975	0.628	0.825	0.957	0.975	0.72	0.866	0.893	0.552	0.898	1	0.989	0.814	0.906
	DAAD+	0.866	0.957	0.862	0.882	0.982	0.976	0.844	0.767	0.921	0.758	0.9	0.987	0.992	0.876	0.859
	MMRR w/o ref.	0.4166	0.981	0.8005	0.9015	0.9842	0.9458	0.8277	0.738	0.9157	0.7085	0.8862	0.5288	0.9816	0.8897	0.8629
	MMRR	0.496	0.9908	0.7993	0.7652	0.9316	0.9595	0.8639	0.7535	0.9107	0.8162	0.8775	0.66	0.9798	0.9162	0.8703

Table 7: Pixel-level and Image-level anomaly detection on MVTecAD dataset.

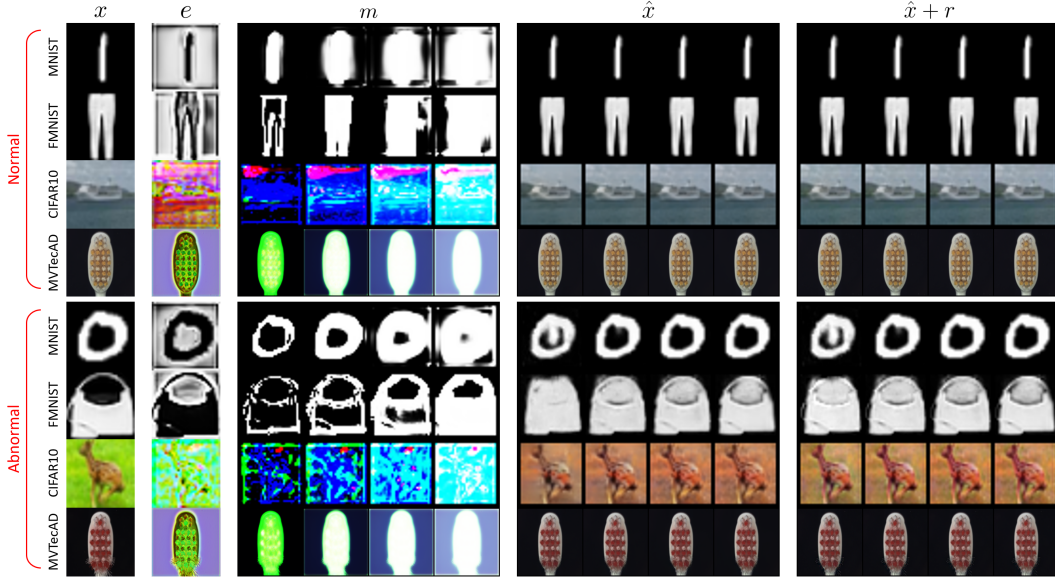


Figure 3: Qualitative results for normal and abnormal samples.

5 Conclusion

We proposed Multi-Level Masking and Restoration with Refinement (MMRR), which started from the motivation to perform anomaly detection through a series of processes of information limitation and restoration. The most noteworthy point of this study is that it presented the *hyperparameter sensitivity* problem for the first time, a problem that had been overlooked in existing anomaly detection studies. MMRR solved the *hyperparameter sensitivity* problem through ensemble at multiple masking levels with novel mask generation method. To empirically demonstrate the robustness to hyperparameter and prior knowledge-free properties of MMRR, we compared the performance as varying the number of masking level and augmentations. Additionally, we solved the problem of not considering the intrinsic complexity of data owing to the novel mask generation method through the refinement module, and achieved comparable performance on MNIST, FMNIST, CIFAR10, and MVTecAD datasets. However, since we have to forward several times for ensemble in multi-level masking, it has the disadvantage of being computationally expensive. We will go further here and try to find a lightweight anomaly detection method without suffering from *hyperparameter sensitivity* problems.

373 **References**

- 374 [1] S. Akçay, A. Atapour-Abarghouei, and T. P. Breckon. Skip-ganomaly: Skip connected and adversarially
375 trained encoder-decoder anomaly detection. In *2019 International Joint Conference on Neural Networks*
376 (*IJCNN*), pages 1–8. IEEE, 2019.
- 377 [2] J. An and S. Cho. Variational autoencoder based anomaly detection using reconstruction probability.
378 *Special Lecture on IE*, 2(1):1–18, 2015.
- 379 [3] L. Bergman and Y. Hoshen. Classification-based anomaly detection for general data. In *International*
380 *Conference on Learning Representations*, 2019.
- 381 [4] L. Bergman, N. Cohen, and Y. Hoshen. Deep nearest neighbor anomaly detection. *arXiv preprint*
382 *arXiv:2002.10445*, 2020.
- 383 [5] P. Bergmann, M. Fauser, D. Sattlegger, and C. Steger. Uninformed students: Student-teacher anomaly
384 detection with discriminative latent embeddings. In *Proceedings of the IEEE/CVF Conference on Computer*
385 *Vision and Pattern Recognition*, pages 4183–4192, 2020.
- 386 [6] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton. A simple framework for contrastive learning of visual
387 representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.
- 388 [7] R. A. Davis, K.-S. Lii, and D. N. Politis. Remarks on some nonparametric estimates of a density function.
389 In *Selected Works of Murray Rosenblatt*, pages 95–100. Springer, 2011.
- 390 [8] L. Deecke, R. Vandermeulen, L. Ruff, S. Mandt, and M. Kloft. Image anomaly detection with generative
391 adversarial networks. In *Joint european conference on machine learning and knowledge discovery in*
392 *databases*, pages 3–17. Springer, 2018.
- 393 [9] T. Defard, A. Setkov, A. Loesch, and R. Audigier. Padim: a patch distribution modeling framework for
394 anomaly detection and localization. In *International Conference on Pattern Recognition*, pages 475–489.
395 Springer, 2021.
- 396 [10] D. Dehaene, O. Frigo, S. Combrexelle, and P. Eline. Iterative energy-based projection on a normal data
397 manifold for anomaly localization. *CoRR*, abs/2002.03734, 2020. URL [https://arxiv.org/abs/2002.](https://arxiv.org/abs/2002.03734)
398 [03734](https://arxiv.org/abs/2002.03734).
- 399 [11] Y. Fei, C. Huang, C. Jinkun, M. Li, Y. Zhang, and C. Lu. Attribute restoration framework for anomaly
400 detection. *IEEE Transactions on Multimedia*, 2020.
- 401 [12] S. Gidaris, P. Singh, and N. Komodakis. Unsupervised representation learning by predicting image
402 rotations. *arXiv preprint arXiv:1803.07728*, 2018.
- 403 [13] I. Golan and R. El-Yaniv. Deep anomaly detection using geometric transformations. *Advances in neural*
404 *information processing systems*, 31, 2018.
- 405 [14] D. Gong, L. Liu, V. Le, B. Saha, M. R. Mansour, S. Venkatesh, and A. v. d. Hengel. Memorizing
406 normality to detect anomaly: Memory-augmented deep autoencoder for unsupervised anomaly detection.
407 In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1705–1714, 2019.
- 408 [15] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio.
409 Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.
- 410 [16] S. Goyal, A. Raghunathan, M. Jain, H. V. Simhadri, and P. Jain. Drocc: Deep robust one-class classification.
411 In *International Conference on Machine Learning*, pages 3711–3721. PMLR, 2020.
- 412 [17] D. Hendrycks, M. Mazeika, and T. Dietterich. Deep anomaly detection with outlier exposure. In
413 *International Conference on Learning Representations*, 2018.
- 414 [18] D. Hendrycks, M. Mazeika, S. Kadavath, and D. Song. Using self-supervised learning can improve model
415 robustness and uncertainty. *Advances in Neural Information Processing Systems*, 32, 2019.
- 416 [19] H. Hoffmann. Kernel pca for novelty detection. *Pattern recognition*, 40(3):863–874, 2007.
- 417 [20] J. Hou, Y. Zhang, Q. Zhong, D. Xie, S. Pu, and H. Zhou. Divide-and-assemble: Learning block-wise
418 memory for unsupervised anomaly detection. In *Proceedings of the IEEE/CVF International Conference*
419 *on Computer Vision*, pages 8791–8800, 2021.
- 420 [21] W. Hu, M. Wang, Q. Qin, J. Ma, and B. Liu. Hrn: A holistic approach to one class learning. *Advances in*
421 *Neural Information Processing Systems*, 33:19111–19124, 2020.

- 422 [22] A. Krizhevsky, G. Hinton, et al. Learning multiple layers of features from tiny images. 2009.
- 423 [23] G. Kwon, M. Prabhushankar, D. Temel, and G. AlRegib. Backpropagated gradient representations for
424 anomaly detection. In *European Conference on Computer Vision*, pages 206–226. Springer, 2020.
- 425 [24] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition.
426 *Proceedings of the IEEE*, 86(11):2278–2324, 1998.
- 427 [25] C.-L. Li, K. Sohn, J. Yoon, and T. Pfister. Cutpaste: Self-supervised learning for anomaly detection and
428 localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*,
429 pages 9664–9674, 2021.
- 430 [26] F. T. Liu, K. M. Ting, and Z.-H. Zhou. Isolation forest. In *2008 eighth ieee international conference on*
431 *data mining*, pages 413–422. IEEE, 2008.
- 432 [27] L. Liu, H. Jiang, P. He, W. Chen, X. Liu, J. Gao, and J. Han. On the variance of the adaptive learning rate
433 and beyond. *CoRR*, abs/1908.03265, 2019. URL <http://arxiv.org/abs/1908.03265>.
- 434 [28] P. Liznerski, L. Ruff, R. A. Vandermeulen, B. J. Franks, M. Kloft, and K.-R. Müller. Explainable deep
435 one-class classification. *arXiv preprint arXiv:2007.01760*, 2020.
- 436 [29] E. Parzen. On estimation of a probability density function and mode. *The annals of mathematical statistics*,
437 33(3):1065–1076, 1962.
- 438 [30] P. Perera, R. Nallapati, and B. Xiang. Ocgan: One-class novelty detection using gans with constrained
439 latent representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern*
440 *Recognition*, pages 2898–2906, 2019.
- 441 [31] S. Pidhorskyi, R. Almoheisen, and G. Doretto. Generative probabilistic novelty detection with adversarial
442 autoencoders. *Advances in neural information processing systems*, 31, 2018.
- 443 [32] M. Pourreza, B. Mohammadi, M. Khaki, S. Bouindour, H. Snoussi, and M. Sabokrou. G2d: generate to
444 detect anomaly. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*,
445 pages 2003–2012, 2021.
- 446 [33] T. Reiss, N. Cohen, L. Bergman, and Y. Hoshen. Panda: Adapting pretrained features for anomaly
447 detection and segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern*
448 *Recognition*, pages 2806–2814, 2021.
- 449 [34] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation.
450 *CoRR*, abs/1505.04597, 2015. URL <http://arxiv.org/abs/1505.04597>.
- 451 [35] L. Ruff, R. Vandermeulen, N. Goernitz, L. Deecke, S. A. Siddiqui, A. Binder, E. Müller, and M. Kloft.
452 Deep one-class classification. In *International conference on machine learning*, pages 4393–4402. PMLR,
453 2018.
- 454 [36] M. Sabokrou, M. Khalooei, M. Fathy, and E. Adeli. Adversarially learned one-class classifier for novelty
455 detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages
456 3379–3388, 2018.
- 457 [37] M. Sakurada and T. Yairi. Anomaly detection using autoencoders with nonlinear dimensionality reduction.
458 In *Proceedings of the MLSDA 2014 2nd workshop on machine learning for sensory data analysis*, pages
459 4–11, 2014.
- 460 [38] M. Salehi, N. Sadjadi, S. Baselizadeh, M. H. Rohban, and H. R. Rabiee. Multiresolution knowledge
461 distillation for anomaly detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and*
462 *Pattern Recognition*, pages 14902–14912, 2021.
- 463 [39] T. Schlegl, P. Seeböck, S. M. Waldstein, U. Schmidt-Erfurth, and G. Langs. Unsupervised anomaly
464 detection with generative adversarial networks to guide marker discovery. In *International conference on*
465 *information processing in medical imaging*, pages 146–157. Springer, 2017.
- 466 [40] B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson. Estimating the support of a
467 high-dimensional distribution. *Neural computation*, 13(7):1443–1471, 2001.
- 468 [41] J. Tack, S. Mo, J. Jeong, and J. Shin. Csi: Novelty detection via contrastive learning on distributionally
469 shifted instances. *Advances in neural information processing systems*, 33:11839–11852, 2020.
- 470 [42] D. M. Tax and R. P. Duin. Support vector data description. *Machine learning*, 54(1):45–66, 2004.

- 471 [43] S. Venkataramanan, K.-C. Peng, R. V. Singh, and A. Mahalanobis. Attention guided anomaly localization
472 in images. In *European Conference on Computer Vision*, pages 485–503. Springer, 2020.
- 473 [44] H. Xiao, K. Rasul, and R. Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine
474 learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.
- 475 [45] J. Yi and S. Yoon. Patch svdd: Patch-level svdd for anomaly detection and segmentation. In *Proceedings
476 of the Asian Conference on Computer Vision*, 2020.
- 477 [46] S. Zagoruyko and N. Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016.
- 478 [47] V. Zavrtnik, M. Kristan, and D. Skočaj. Draem-a discriminatively trained reconstruction embedding for
479 surface anomaly detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*,
480 pages 8330–8339, 2021.
- 481 [48] V. Zavrtnik, M. Kristan, and D. Skočaj. Reconstruction by inpainting for visual anomaly detection.
482 *Pattern Recognition*, 112:107706, 2021.

483 Checklist

484 The checklist follows the references. Please read the checklist guidelines carefully for information on
485 how to answer these questions. For each question, change the default **[TODO]** to **[Yes]**, **[No]**, or
486 **[N/A]**. You are strongly encouraged to include a **justification to your answer**, either by referencing
487 the appropriate section of your paper or providing a brief inline description. For example:

- 488 • Did you include the license to the code and datasets? **[Yes]** See Section ??.
- 489 • Did you include the license to the code and datasets? **[No]** The code and the data are proprietary.
- 490 • Did you include the license to the code and datasets? **[N/A]**

491 Please do not modify the questions and only use the provided macros for your answers. Note that the
492 Checklist section does not count towards the page limit. In your paper, please delete this instructions
493 block and only keep the Checklist section heading above along with the questions/answers below.

494 1. For all authors...

- 495 (a) Do the main claims made in the abstract and introduction accurately reflect the paper’s
496 contributions and scope? **[Yes]** The contributions of our work are thoroughly are
497 included in both abstract and introduction.
- 498 (b) Did you describe the limitations of your work? **[Yes]** The limitations of our work are
499 mentioned in Section 5.
- 500 (c) Did you discuss any potential negative societal impacts of your work? **[N/A]**
- 501 (d) Have you read the ethics review guidelines and ensured that your paper conforms to
502 them? **[Yes]**

503 2. If you are including theoretical results...

- 504 (a) Did you state the full set of assumptions of all theoretical results? **[N/A]**
- 505 (b) Did you include complete proofs of all theoretical results? **[N/A]**

506 3. If you ran experiments...

- 507 (a) Did you include the code, data, and instructions needed to reproduce the main experi-
508 mental results (either in the supplemental material or as a URL)? **[Yes]** We include all
509 the information required to reproduce our experimental results in Section 4.
- 510 (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they
511 were chosen)? **[Yes]** All the training details are mentioned in Section 4.
- 512 (c) Did you report error bars (e.g., with respect to the random seed after running experi-
513 ments multiple times)? **[Yes]** We reported error bars in the tables in Section 4.
- 514 (d) Did you include the total amount of compute and the type of resources used (e.g., type
515 of GPUs, internal cluster, or cloud provider)? **[Yes]** They are reported in Appendix.

516 4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...

- 517 (a) If your work uses existing assets, did you cite the creators? [Yes] We cited all the
518 creators for the assets we use throughout our paper.
- 519 (b) Did you mention the license of the assets? [N/A]
- 520 (c) Did you include any new assets either in the supplemental material or as a URL? [No]
- 521 (d) Did you discuss whether and how consent was obtained from people whose data you're
522 using/curating? [Yes] We use the data widely used in the field related to our work.
- 523 (e) Did you discuss whether the data you are using/curating contains personally identifiable
524 information or offensive content? [N/A]
- 525 5. If you used crowdsourcing or conducted research with human subjects...
- 526 (a) Did you include the full text of instructions given to participants and screenshots, if
527 applicable? [N/A]
- 528 (b) Did you describe any potential participant risks, with links to Institutional Review
529 Board (IRB) approvals, if applicable? [N/A]
- 530 (c) Did you include the estimated hourly wage paid to participants and the total amount
531 spent on participant compensation? [N/A]