Text-Guided Variational Image Generation for Industrial Anomaly Detection and Segmentation

Supplementary Material

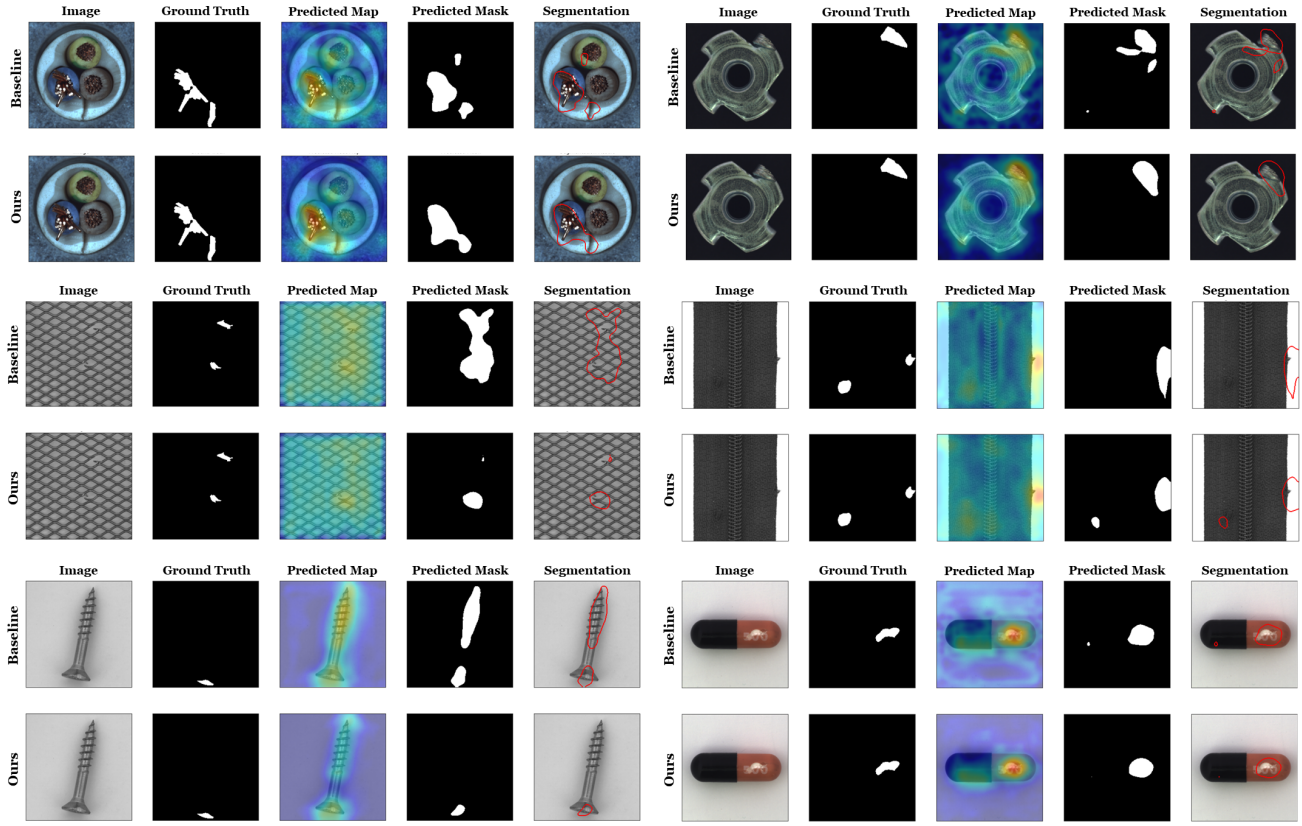

Figure 11. **Comparison of qualitative results between baseline(Patchcore) and Ours in MVTecAD dataset**. Using our model, we show better detection and segmentation performance for each class compared to the baseline.

A. Appendix

Here, we present additional findings that, due to space limitations, could not be included in the main paper. First, detailed implementation, including evaluation metrics and baselines, benchmark datasets, were provided. In addition, the comprehensive quantitative result (Table 5), overall classes performance results by baseline (Tables 6 $\sim$ 15) and qualitative experimental results (Figs. 11, 13 $\sim$ 14). Additional analysis results and discussion were also provided.

A.1. Comparison of qualitative results in MVTecAD dataset.

In Fig. 11, we can visually confirm that our method is improved over the baseline through the qualitative results of few classes. In Table 4 are few examples of prompts that were finally selected and applied.

Table 4. **Prompt examples by Keyword-to-Prompt Generator**.

Best prompt	Worst prompt
"a {hazelnut} with {cobnut}"	"a {hazelnut} with {decantherous}"
"a {metalnut} with {metallical}"	"a {metalnut} with {predegenerate}"
"a {zipper} with {metallization}"	"a {zipper} with {Echinops}"
"a {capsule} with {incapsulation}"	"a {capsule} with {perceptible }"
"a {toothbrush} with {parazoan}"	"a {toothbrush} with {chaetopod}"

A.2. Implementation Details

The basic settings of each module are as follows:

The variance-aware image generator is initialized by the weight parameters of VQGAN [10] pre-trained by ImageNet. In the Keyword-prompt generator, the number of candidate prompts $(S_1, S_2, \ldots, S_T)$ using WordNet [11] is 1,000; among the candidate prompts, the one closest to the input image is selected by 100 times iteration. The text-guided knowledge integrator, the encoder, uses a pre-trained CLIP model [19] based on 'ViT-B/16'.

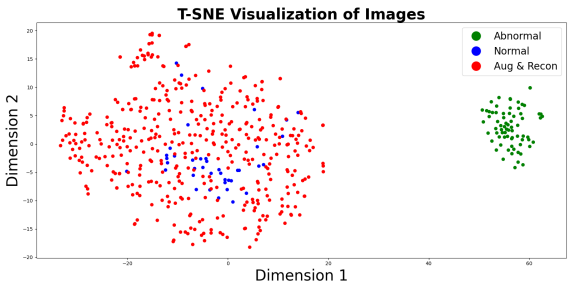

Figure 12. **Visual analysis on t-SNE distributions.** (Patchcore, Toothbrush(class)) We compare t-SNE distributions with the original images (Non-defective), generated images (Aug & Recon), and defective images (Defective). The graph shows that the generated images are generalized well with non-defective images, while effectively distinguishing the abnormal samples.

The experimental setup involves using the official code provided by the authors for each baseline, with results averaged over five runs. Each experiment was repeated 20 times with a constant learning rate of 0.05. VRAM usage during the experiments varied between 2,031 $\sim$ 23,277(MB), with an average runtime of 182.6(sec) per iteration. In all scenarios (one-shot, few-shot, and full-shot), baseline configurations were used as specified by their respective authors, with ResNet-18 serving as the default backbone network. And for the reverse-distillation model, a WideResNet50-2 was utilized. The AUROC metric was employed to evaluate both detection and segmentation performances.

A.2.1 Evaluation Metrics.

The AUROC (Area Under the Receiver Operating Characteristic curve) is a critical metric in classification models, particularly for binary classification scenarios. It measures a model's ability to differentiate between two distinct classes accurately. This metric is depicted as the area beneath the ROC curve, representing the True Positive Rate(TPR) versus the False Positive Rate(FPR) across a spectrum of threshold settings. The AUROC value, which can vary between 0 and 1, directly correlates with the model's discriminate ability; higher values indicate superior performance.

In the specific context of Anomaly Detection and Segmentation, performance evaluation extends beyond conventional metrics to include specialized tests such as the one-shot, few-shot, and full-shot tests. The one-shot test evaluates model performance using a single non-defective image and its generated counterparts, assessing its ability to detect anomalies based on minimal data. The few-shot test enhances this approach by using five non-defective images as input, providing a slightly broader base for generating and evaluating images. On the other hand, the full-shot test employs the entire dataset of non-defective images for training and image generation, offering a comprehensive evaluation of the model's capability to identify and segment anomalies.

A.2.2 Baselines and Comparisons

We perform a comparative analysis for four baselines. The Patchcore [21] algorithm extracts features with the Coreset sampling module to use only a part of the training data. The Cflow [14] uses a normalizing flow to directly predict the test image's probability by learning the invertible function mapping from images to Gaussian distribution. The Efficient-AD [2] uses a student-teacher approach to detect anomalous features. The authors of Efficient-AD train a student network to predict the extracted features of non-defective data, i.e., anomaly-free training images. The student's failure to predict their features is considered a detection of anomalies. The Reverse Distillation [9] proposes a teacher encoder, student decoder, and reverse distillation paradigm model. Instead of receiving raw images directly, the student network takes the teacher model's one-class embedding as input and targets to restore the teacher's multi-scale representations.

A.2.3 Benchmark Datasets

We conducted a comparative analysis using three datasets to compare the models' effectiveness. The MVTecAD (MVTec Anomaly Detection) [3] dataset is a benchmark dataset for anomaly detection methods focusing on industrial inspection. It contains over 5,000 high-resolution images divided into fifteen object and texture categories. Each category comprises a set of defect-free training images, a test set of images with various defects, and images without defects. The BTAD (beanTech Anomaly Detection) [18] dataset is a real-world industrial anomaly dataset containing 2,830 real-world images of 3 industrial products containing body and surface defects. The MVTec-LOCO AD [4] dataset is intended to evaluate unsupervised anomaly localization algorithms. It includes structural and logical anomalies, containing 3,644 images from five categories of real-world industrial inspection scenarios. The structural anomalies have scratches, dents, and contaminations in the manufactured products. The logical anomalies violate the underlying constraints, such as an invalid location and a missing object. The dataset also includes pixel-precise ground truth data for the segmentation task.

A.3. Detailed Experimental Result

A.3.1 Comprehensive Comparison

In Table 5, our method shows improved performance compared to all baselines. In particular, good performance can be obtained even under limited conditions (one- or few-shot scenarios), indicating that the model effectively incorporates features of non-defective images into the images generated by our method.

Table 5. **Comprehensive results on MVTecAD(All baselines, All classes, Detection/Segmentation AUROC)**, Experimental results show improved performance in all scenarios. All experimental results were measured with the official code provided by the author for each baseline and calculated as the average value performed five times.

Baseline	One-shot	Few-shot	Full-shot
Patchcore [21]	76.9% / 90.8%	85.8% / 93.2%	97.3% / 97.6%
Ours	83.9% / 92.8%	91.4% / 95.1%	97.8% / 97.7%
Gain(+%)	**+7.0% / +2.0%**	**+5.6% / +1.9%**	**+0.5% / +0.1%**
Cflow [14]	69.0% / 89.0%	79.8% / 92.8%	88.8% / 96.1%
Ours	84.2% / 91.6%	90.2% / 94.9%	93.7% / 95.9%
Gain(+%)	**+15.2% / +2.6%**	**+10.4% / +2.1%**	**+4.9% / -0.2%**
Reverse-distillation [9]	48.4% / 29.4%	50.6% / 32.7%	79.2% / 97.0%
Ours	83.1% / 93.3%	84.2% / 92.1%	93.3% / 97.1%
Gain(+%)	**+34.7% / +63.9%**	**+33.7% / +59.4%**	**+14.1% / +0.1%**
Efficient-AD [2]	65.6% / 78.1%	71.3% / 83.3%	92.8% / 93.6%
Ours	71.3% / 83.7%	78.1% / 85.9%	95.5% / 94.3%
Gain(+%)	**+5.8% / +5.6%**	**+6.8% / +2.6%**	**+2.7% / +0.7%**
Average	**+15.7% / +18.6%**	**+14.1% / +16.5%**	**+5.6% / +0.2%**

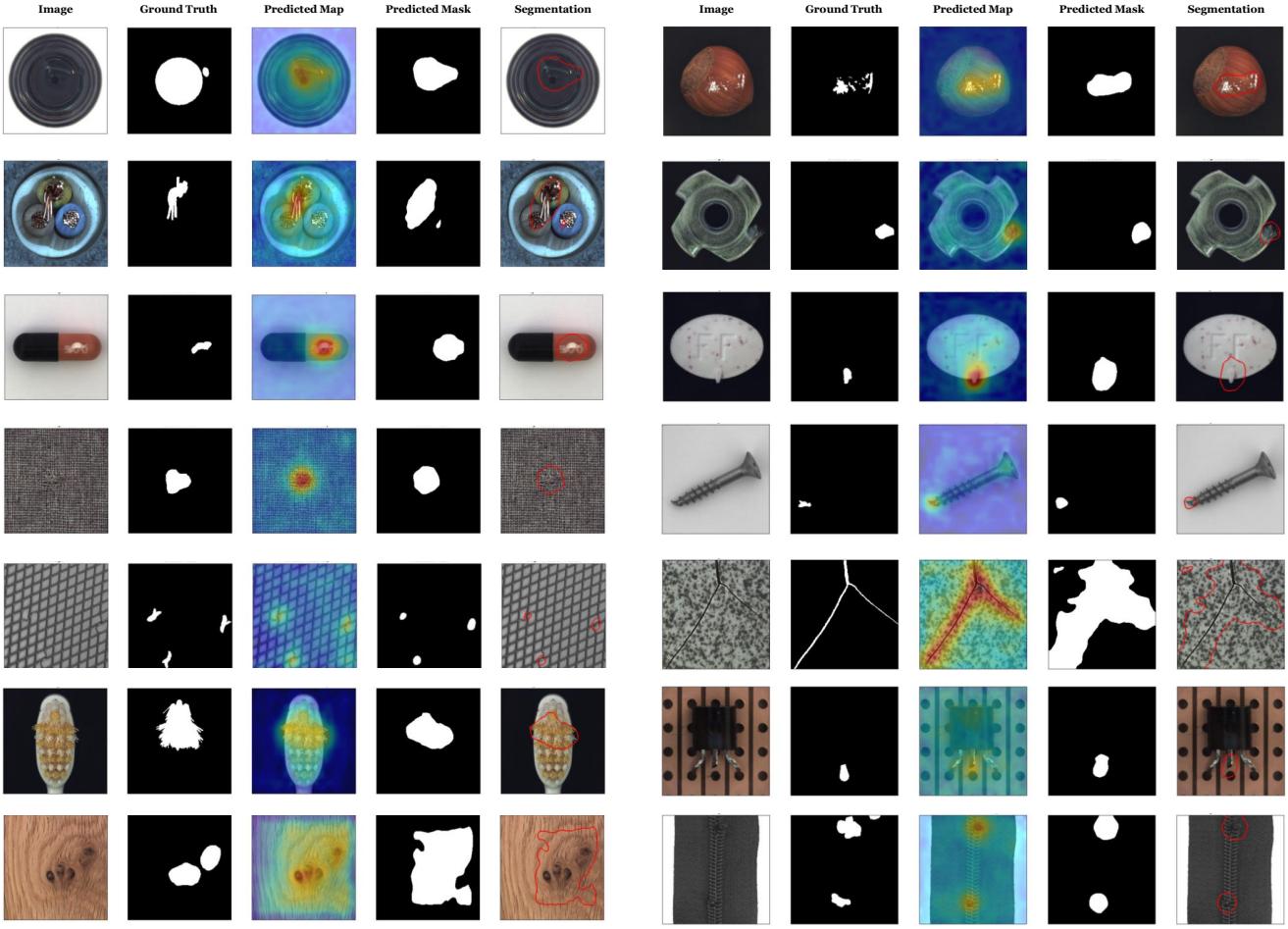

Figure 13. **Qualitative results[All classes] on MVTecAD dataset**. For each class in the scenario test, we show the qualitative results of defective images using our model.

Table 6. **Generalization test on MVTecAD(Patchcore, Object type classes, Detection AUROC)**, Generalization test on MVTec-AD dataset [3], and the values of object type among all classes in the patchcore [21] baseline and their average values are expressed. All experimental results were measured using the official code provided by the author for each baseline and calculated as the average value performed five times.

	Bottle	Cable	Capsule	Hazelnut	Metal-nut	Pill	Screw	Toothbrush	Transistor	Zipper	Avg
One-shot	99.5(±0.2)	70.8(±1.1)	55.5(±1.6)	88.3(±0.2)	70.3(±0.7)	69.2(±2.4)	48.1(±1.4)	74.7(±0.7)	65.3(±3.0)	78.3(±1.2)	72.0
Ours	100.0(±0.0)	89.9(±0.3)	63.6(±0.2)	91.7(±0.1)	88.8(±0.2)	72.0(±0.4)	53.8(±0.1)	78.7(±0.4)	71.8(±0.7)	91.5(±0.2)	**80.2**
Gain(+%)	**+0.5%**	**+19.1%**	**+8.1%**	**+3.4%**	**+18.5%**	**+2.8%**	**+5.7%**	**+4.0%**	**+6.5%**	**+13.2%**	**+8.2%**
Few-shot	99.3(±0.1)	87.7(±0.9)	77.7(±1.8)	96.0(±1.1)	95.3(±0.4)	84.3(±2.7)	50.3(±0.3)	63.3(±0.7)	93.5(±0.3)	91.9(±0.3)	83.9
Ours	100.0(±0.0)	95.6(±0.8)	93.0(±0.4)	99.3(±0.1)	99.1(±0.1)	88.9(±4.3)	66.6(±0.3)	69.5(±0.2)	98.9(±0.2)	92.4(±0.1)	**90.3**
Gain(+%)	**+0.7%**	**+7.9%**	**+15.3%**	**+3.3%**	**+3.8%**	**+4.6%**	**+16.3%**	**+6.2%**	**+5.4%**	**+0.5%**	**+6.4%**
Full-shot	100.0(±0.0)	98.6(±0.1)	96.5(±0.6)	100.0(±0.0)	99.1(±0.3)	91.6(±1.5)	94.3(±0.6)	93.1(±1.1)	99.6(±0.2)	95.3(±0.1)	96.8
Ours	100.0(±0.0)	98.8(±0.1)	97.2(±0.3)	100.0(±0.0)	100.0(±0.0)	92.5(±0.2)	96.1(±0.5)	95.7(±0.9)	100.0(±0.0)	95.7(±0.1)	**97.6**
Gain(+%)	**+0.0%**	**+0.2%**	**+0.7%**	**+0.0%**	**+0.9%**	**+0.9%**	**+1.8%**	**+2.6%**	**+0.4%**	**+0.4%**	**+0.8%**

Table 7. **Generalization test on MVTecAD(Patchcore, Texture type classes, Detection AUROC)**, Generalization test on MVTec-AD dataset [3], and the values of texture type among all classes in the patchcore [21] baseline and their average values are expressed. All experimental results were measured using the official code provided by the author for each baseline and calculated as the average value performed five times.

	Carpet	Grid	Leather	Tile	Wood	Avg
One-shot	89.4(±2.2)	47.2(±0.2)	99.9(±0.1)	99.3(±0.2)	97.6(±0.3)	86.7
Ours	93.6(±0.5)	64.9(±0.2)	100.0(±0.0)	99.3(±0.1)	98.9(±0.1)	**91.3**
Gain(+%)	**+4.2%**	**+17.7%**	**+0.1%**	**+0.0%**	**+1.3%**	**+4.6%**
Few-shot	93.4(±0.7)	56.4(±2.9)	100.0(±0.0)	99.4(±0.2)	98.7(±0.1)	89.6
Ours	96.4(±0.3)	72.3(±0.8)	100.0(±0.0)	99.8(±0.0)	99.5(±0.0)	**93.6**
Gain(+%)	**+3.0%**	**+15.9%**	**+0.0%**	**+0.4%**	**+0.8%**	**+4.0%**
Full-shot	97.0(±0.5)	94.7(±0.4)	100.0(±0.0)	99.7(±0.1)	99.7(±0.1)	98.2
Ours	96.7(±0.2)	94.3(±0.4)	100.0(±0.0)	99.9(±0.0)	99.7(±0.0)	**98.1**
Gain(+%)	**-0.3%**	**-0.4%**	**+0.0%**	**+0.2%**	**+0.0%**	**-0.1%**

Table 8. **Generalization test on MVTecAD(Cflow, Object type classes, Detection AUROC)**, Generalizationtion test on MVTec-AD dataset [3], and the values of Object type among all classes in the cflow [14] baseline and their average values are expressed. All experimental results were measured using the official code provided by the author for each baseline and calculated as the average value performed five times.

	Bottle	Cable	Capsule	Hazelnut	Metal-nut	Pill	Screw	Toothbrush	Transistor	Zipper	Avg
One-shot	93.7(±1.1)	57.8(±6.9)	71.5(±0.7)	90.5(±0.7)	60.1(±1.4)	68.6(±3.2)	54.6(±1.1)	68.3(±2.3)	61.8(±3.0)	52.3(±3.4)	67.9
Ours	99.1(±0.9)	77.7(±1.9)	73.6(±3.3)	96.4(±1.3)	79.6(±3.0)	61.0(±6.5)	79.6(±6.4)	71.1(±1.5)	73.7(±5.1)	92.1(±1.5)	**80.4**
Gain(+%)	**+5.4%**	**+19.9%**	**+2.1%**	**+5.9%**	**+19.5%**	**-7.6%**	**+25.0%**	**+2.8%**	**+11.9%**	**+39.8%**	**+12.5%**
Few-shot	96.2(±0.3)	79.9(±1.3)	68.7(±1.4)	96.4(±0.6)	75.8(±2.9)	73.6(±4.1)	50.4(±3.6)	82.3(±3.9)	68.2(±3.1)	73.3(±3.9)	76.5
Ours	99.9(±0.1)	92.5(±0.9)	77.6(±5.1)	99.5(±0.3)	92.1(±2.1)	88.0(±3.1)	61.7(±7.5)	93.9(±1.6)	86.5(±1.6)	92.5(±0.7)	**88.4**
Gain(+%)	**+3.7%**	**+12.6%**	**+8.9%**	**+3.1%**	**+16.3%**	**+14.4%**	**+11.3%**	**+11.6%**	**+18.3%**	**+19.2%**	**+11.9%**
Full-shot	100.0(±0.0)	89.4(±1.9)	80.6(±1.9)	99.1(±0.6)	97.9(±0.6)	84.2(±5.8)	55.6(±4.3)	87.7(±1.3)	84.0(±0.9)	91.9(±0.6)	87.0
Ours	100.0(±0.0)	95.7(±0.9)	91.6(±1.6)	99.9(±0.1)	99.2(±0.4)	91.2(±1.2)	74.9(±4.0)	91.6(±1.3)	91.2(±4.8)	95.0(±0.9)	**93.0**
Gain(+%)	**+0.0%**	**+6.3%**	**+11.0%**	**+0.8%**	**+1.3%**	**+7.0%**	**+19.3%**	**+3.9%**	**+7.2%**	**+3.1%**	**+6.0%**

Table 9. **Generalization test on MVTEC-AD(Cflow, Texture type classes, Detection AUROC)**, Generalizationtion test on MVTec-AD dataset [3], and the values of texture type among all classes in the cflow [14] baseline and their average values are expressed. All experimental results were measured using the official code provided by the author for each baseline and calculated as the average value performed five times.

	Carpet	Grid	Leather	Tile	Wood	Avg
One-shot	88.2(±4.3)	33.9(±1.8)	92.7(±0.7)	94.0(±0.6)	96.2(±2.1)	81.0
Ours	96.8(±1.4)	53.8(±11.4)	94.6(±2.3)	96.5(±0.7)	99.0(±0.2)	**88.1**
Gain(+%)	**+8.6%**	**+19.9%**	**+1.9%**	**+2.5%**	**+2.8%**	**+7.1%**
Few-shot	91.5(±1.6)	55.5(±0.9)	95.5(±1.3)	96.0(±0.6)	93.7(±3.0)	86.4
Ours	95.8(±0.7)	77.7(±2.3)	99.1(±0.5)	98.3(±0.6)	97.9(±0.5)	**93.8**
Gain(+%)	**+4.3%**	**+22.2%**	**+3.6%**	**+2.3%**	**+4.2%**	**+7.4%**
Full-shot	94.1(±0.4)	76.4(±1.9)	98.0(±0.4)	96.0(±2.8)	96.8(±2.0)	92.3
Ours	95.2(±0.8)	84.1(±4.2)	98.6(±0.3)	98.2(±0.7)	98.8(±0.4)	**95.0**
Gain(+%)	**+1.1%**	**+7.7%**	**+0.6%**	**+2.2%**	**+2.0%**	**+2.7%**

Table 10. **Generalization test on MVTecAD(Reverse-distillation, Object type classes, Detection AUROC)**, Generalization test on MVTec-AD dataset [3], and the values of object type among all classes in the reverse-distillation [9] baseline and their average values are expressed. All experimental results were measured using the official code provided by the author for each baseline and calculated as the average value performed five times.

	Bottle	Cable	Capsule	Hazelnut	Metal-nut	Pill	Screw	Toothbrush	Transistor	Zipper	Avg
One-shot	44.1(±0.5)	47.9(±0.1)	58.8(±0.0)	37.2(±0.2)	45.4(±12.7)	57.1(±0.3)	53.7(±1.8)	39.0(±0.2)	55.1(±7.1)	41.9(±1.0)	48.0
Ours	98.0(±1.8)	65.0(±1.0)	67.4(±2.2)	99.9(±0.1)	62.7(±2.1)	81.2(±0.9)	54.0(±1.2)	92.0(±0.6)	70.4(±0.8)	78.0(±1.1)	**76.9**
Gain(+%)	**+53.9%**	**+17.1%**	**+8.6%**	**+62.7%**	**+17.3%**	**+24.1%**	**+0.3%**	**+53.0%**	**+15.3%**	**+36.1%**	**+28.9%**
Few-shot	55.2(±0.0)	54.1(±0.0)	53.2(±0.0)	35.7(±0.0)	40.7(±0.0)	53.1(±0.0)	54.7(±0.0)	44.7(±0.0)	60.5(±14.6)	34.6(±0.0)	48.6
Ours	98.6(±1.5)	84.6(±3.3)	74.7(±1.0)	100.0(±0.0)	79.9(±11.8)	79.8(±0.7)	63.6(±0.9)	94.2(±0.4)	84.0(±2.2)	73.7(±0.7)	**83.3**
Gain(+%)	**+43.4%**	**+30.5%**	**+21.5%**	**+64.3%**	**+39.2%**	**+26.7%**	**+8.9%**	**+49.5%**	**+23.5%**	**+39.1%**	**+34.7%**
Full-shot	95.3(±3.8)	95.0(±0.7)	88.5(±4.0)	100.0(±0.0)	61.0(±2.4)	59.4(±0.5)	93.4(±1.7)	75.9(±10.8)	80.1(±7.9)	89.8(±0.9)	83.8
Ours	98.7(±1.8)	93.9(±1.7)	94.4(±0.7)	100.0(±0.0)	75.1(±17.8)	95.4(±2.5)	95.7(±1.3)	96.6(±0.6)	97.1(±0.6)	92.1(±1.7)	**93.9**
Gain(+%)	**+3.4%**	**-1.1%**	**+5.9%**	**+0.0%**	**+14.1%**	**+36.0%**	**+2.3%**	**+20.7%**	**+17.0%**	**+2.3%**	**+10.1%**

Table 11. **Generalization test on MVTecAD(Reverse-distillation, Texture type classes, Detection AUROC)**, Generalization test on MVTec-AD dataset [3], and the values of texture type among all classes in the reverse-distillation [9] baseline and their average values are expressed. All experimental results were measured using the official code provided by the author for each baseline and calculated as the average value performed five times.

	Carpet	Grid	Leather	Tile	Wood	Avg
One-shot	63.0(±32.9)	56.9(±0.1)	26.7(±0.2)	45.7(±0.5)	54.3(±0.5)	49.3
Ours	99.4(±0.0)	79.7(±6.7)	100.0(±0.0)	99.4(±0.1)	99.6(±0.0)	**95.6**
Gain(+%)	**+36.4%**	**+22.8%**	**+73.3%**	**+53.7%**	**+45.3%**	**+46.3%**
Few-shot	35.3(±0.0)	52.1(±0.0)	42.4(±1.0)	23.4(±0.0)	53.3(±0.0)	41.3
Ours	99.4(±0.1)	85.1(±1.7)	99.9(±0.0)	98.0(±0.3)	99.8(±0.1)	**96.4**
Gain(+%)	**+64.1%**	**+33.0%**	**+57.5%**	**+74.6%**	**+46.5%**	**+55.1%**
Full-shot	99.4(±0.0)	96.6(±1.6)	39.6(±0.5)	78.0(±23.0)	99.7(±0.0)	82.7
Ours	99.5(±0.1)	99.5(±0.2)	100.0(±0.0)	99.6(±0.1)	99.7(±0.0)	**99.7**
Gain(+%)	**+0.1%**	**+2.9%**	**+60.4%**	**+21.6%**	**+0.0%**	**+17.0%**

Table 12. **Generalization test on MVTecAD(Efficient-AD, Object type classes, Detection AUROC)**, Generalization test on MVTec-AD dataset [3], and the values of object type among all classes in the Efficient-AD [2] baseline and their average values are expressed. All experimental results were measured using the official code provided by the author for each baseline and calculated as the average value performed five times.

	Bottle	Cable	Capsule	Hazelnut	Metal-nut	Pill	Screw	Toothbrush	Transistor	Zipper	Avg
One-shot	84.0(±0.0)	50.2(±0.1)	39.6(±0.0)	72.8(±0.0)	37.4(±0.0)	69.6(±0.1)	66.4(±0.0)	42.8(±0.1)	34.9(±0.0)	56.3(±0.0)	55.4
Ours	95.1(±0.7)	61.9(±0.7)	51.7(±0.6)	75.9(±0.8)	62.1(±0.6)	73.2(±2.3)	74.3(±1.1)	63.4(±1.7)	58.5(±1.3)	52.8(±0.2)	**66.9**
Gain(+%)	**+11.1%**	**+11.7%**	**+12.1%**	**+3.1%**	**+24.7%**	**+3.6%**	**+7.9%**	**+20.6%**	**+23.6%**	**-3.5%**	**+11.5%**
Few-shot	96.2(±0.1)	60.5(±0.2)	45.2(±0.3)	69.5(±0.2)	52.6(±0.4)	70.4(±0.7)	63.8(±0.1)	51.2(±0.4)	43.3(±0.1)	54.9(±0.1)	60.8
Ours	98.0(±0.1)	75.9(±0.4)	55.2(±0.2)	87.3(±0.7)	67.9(±1.9)	75.1(±1.0)	75.4(±3.7)	69.2(±0.9)	60.8(±1.4)	57.6(±1.5)	**72.2**
Gain(+%)	**+1.8%**	**+15.4%**	**+10.0%**	**+17.8%**	**+15.3%**	**+4.7%**	**+11.6%**	**+18.0%**	**+17.5%**	**+2.7%**	**+11.4%**
Full-shot	100.0(±0.0)	91.9(±0.2)	76.2(±1.2)	89.1(±0.8)	96.7(±0.1)	95.5(±1.0)	90.3(±0.6)	94.8(±0.5)	82.4(±2.1)	94.6(±0.4)	91.2
Ours	100.0(±0.0)	94.7(±0.3)	80.7(±2.6)	94.6(±0.8)	97.0(±0.1)	97.5(±0.6)	95.1(±0.5)	89.3(±0.8)	82.5(±3.6)	94.4(±0.8)	**92.6**
Gain(+%)	**+0.0%**	**+2.8%**	**+4.5%**	**+5.5%**	**+0.3%**	**+2.0%**	**+4.8%**	**-5.5%**	**+0.1%**	**-0.2%**	**+1.4%**

Table 13. **Generalization test on MVTecAD(Efficient-AD, Texture type classes, Detection AUROC)**, Generalization test on MVTec-AD dataset [3], and the values of texture type among all classes in the Efficient-AD [2] baseline and their average values are expressed. All experimental results were measured using the official code provided by the author for each baseline and calculated as the average value performed five times.

	Carpet	Grid	Leather	Tile	Wood	Avg
One-shot	99.4(±0.0)	83.2(±0.0)	65.8(±0.1)	95.2(±0.0)	85.7(±0.0)	85.9
Ours	99.5(±0.1)	74.3(±4.8)	54.8(±8.6)	97.4(±0.1)	81.4(±0.9)	**81.5**
Gain(+%)	**+0.1%**	**-8.9%**	**-11.0%**	**+2.2%**	**-4.3%**	**-4.4%**
Few-shot	98.4(±0.1)	94.2(±0.1)	91.4(±4.7)	94.3(±0.2)	82.9(±0.4)	92.2
Ours	97.8(±0.3)	97.9(±0.4)	66.3(±1.2)	96.2(±0.9)	90.7(±0.5)	**89.8**
Gain(+%)	**-0.6%**	**+3.7%**	**-25.1%**	**+1.9%**	**+7.8%**	**-2.4%**
Full-shot	99.1(±0.2)	99.5(±0.2)	97.5(±0.3)	99.8(±0.0)	96.7(±0.5)	98.5
Ours	98.3(±0.5)	99.4(±0.1)	97.3(±0.2)	99.9(±0.1)	95.6(±0.3)	**98.1**
Gain(+%)	**-0.8%**	**-0.1%**	**-0.2%**	**+0.1%**	**-1.1%**	**-0.4%**

Table 14. **Generalization test on BTAD(Patchcore, All classes, Detection AUROC)**, As a generalization test for BTAD dataset [18], all class values of the patchcore [21] baseline and their average values were expressed. In the case of Class 1 and Class 3, it shows the object form, and in the case of Class 2, it shows the texture form. All experimental results were measured using the official code provided by the author for each baseline and calculated as the average value performed five times.

	Class-1	Class-2	Class-3	Avg
One-shot	72.2(±7.2)	73.3(±1.7)	64.6(±1.7)	70.0
Ours	92.4(±0.3)	77.3(±0.4)	70.5(±0.8)	**80.1**
Gain(+%)	**+20.2%**	**+4.0%**	**+5.9%**	**+10.1%**
Few-shot	91.7(±0.7)	80.5(±0.7)	67.7(±2.4)	80.0
Ours	94.4(±0.3)	80.7(±0.8)	68.3(±1.0)	**81.1**
Gain(+%)	**+2.7%**	**+0.2%**	**+0.6%**	**+1.1%**
Full-shot	94.3(±0.5)	81.8(±0.6)	68.6(±1.1)	81.6
Ours	94.1(±0.9)	82.4(±0.7)	67.5(±0.5)	**81.3**
Gain(+%)	**-0.2%**	**+0.6%**	**-1.1%**	**-0.3%**

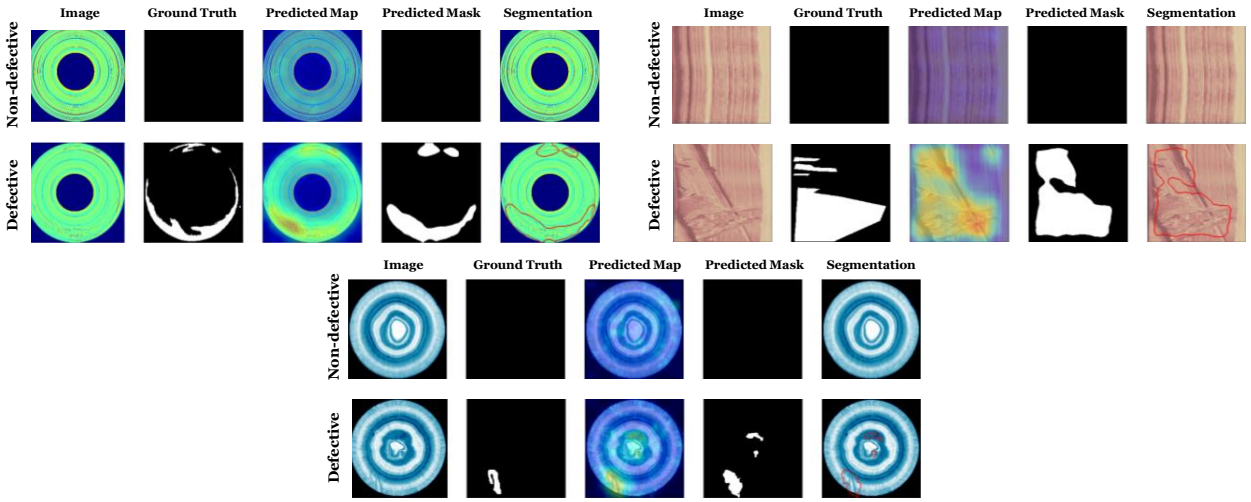

Figure 14. **Qualitative results[All classes] on BTAD dataset**. As shown in the figure, our model obtained qualitative results between non-defective and defect images for the three classes.

Table 15. **Generalization test on MVTec-LOCO AD(Patchcore, All classes, Detection AUROC)**, As a generalization test for MVTec-LOCO AD dataset [4], all class values of the patchcore [21] baseline, and their average values include logical anomalies and structural anomalies for each class were expressed. All experimental results were measured using the official code provided by the author for each baseline and calculated as the average value performed five times.

	Class-1	Class-2	Class-3	Class-4	Class-5	Avg
One-shot	59.4(±0.2)	60.6(±11.9)	54.1(±2.7)	43.0(±0.5)	62.1(±1.4)	55.8
Ours	74.9(±1.4)	71.2(±6.0)	59.1(±0.5)	46.6(±0.3)	64.0(±0.4)	**63.2**
Gain(+%)	**+15.5%**	**+10.6%**	**+5.0%**	**+3.6%**	**+1.9%**	**+7.4%**
Few-shot	62.3(±1.4)	83.4(±0.8)	54.4(±2.2)	54.6(±0.7)	65.7(±0.5)	64.1
Ours	67.0(±1.3)	91.4(±0.4)	54.8(±0.7)	53.5(±1.2)	70.5(±0.3)	**67.4**
Gain(+%)	**+4.7%**	**+8.0%**	**+0.4%**	**-1.1%**	**+4.8%**	**+3.3%**
Full-shot	79.7(±0.6)	96.0(±0.3)	74.0(±1.0)	62.5(±0.9)	81.8(±0.4)	78.8
Ours	78.7(±1.0)	96.3(±0.2)	72.4(±1.1)	63.7(±1.1)	81.6(±1.0)	**78.5**
Gain(+%)	**-1.0%**	**+0.3%**	**-1.6%**	**+1.2%**	**-0.2%**	**-0.3%**

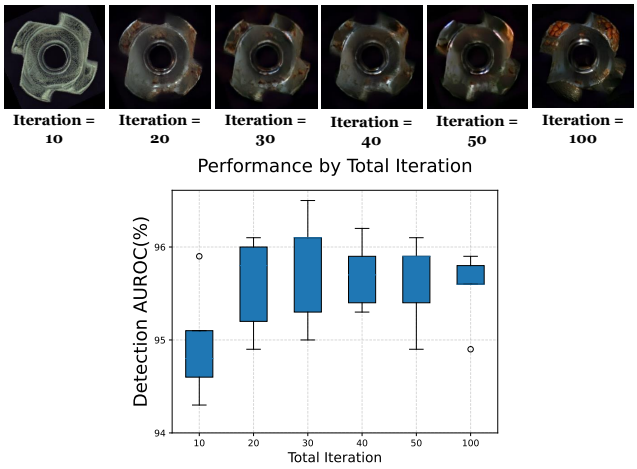

Figure 15. **Comparison results of images generated according to the number of iterations.** (Reverse-distillation, Metal-nut(class)), The generated images(top) show that strongly reflects input text information as repetition increases. The performance results(bottom) show a trend in which performance variance gradually decreases with saturation at a certain point.

A.3.2 Additional Quantitative Results

We show our performance against each baseline, organized across all scenarios (one, few, full-shot) and all classes. First, we compared the average performance by dividing the object type (Tables 6, 8, 10, 12) and texture types (Tables 7, 9, 11, 13). In the Patchcore, Cflow, and Efficient-AD models, the Object Type shows significantly higher performance, but in the reverse distillation model, the Texture Type shows higher performance.

Therefore, we revalidated it on the BTAD dataset (Table 14) in which object type and texture type, and it shows the highest performance in object type. Finally, we tested it on the complex MVTec-LOCO AD dataset (Table 15) and showed performance improvement overall. In particular, the one-shot scenario showed good performance, improving by 15.5% and 10.6% in the breakfast box and juice bottle classes.

A.3.3 Additional Qualitative Results

We show the qualitative results for representative classes, including the test images with ground truth masks for detection and the anomaly localization score heatmap for segmentation. These results can be found in Figs. 13, 14 for the MVTecAD and BTAD datasets, respectively.

A.4. Additional Analysis Results

A.4.1 Visual analysis on t-SNE distributions

Fig. 12 shows the T-SNE distributions to compare the latent features of original non-defective images in blue dots, our generated images in red dots, and defective images in green dots. The results show that the samples generated by our model are evenly distributed close to the non-defective data while effectively separating the defective data.

Table 16. **Ablation studies by Variance-aware parameters.** The result of comparing evaluation metrics (SSIM, PSNR, VIF, LPIPS) that measure the similarity and quality of the original image and a single generated image. The image generated by each parameter of the latent vector distribution in the variance-aware method was compared with the original image.

Method	Avg	SSIM(↑)	PSNR(↑)	VIF(↑)	LPIPS(↓)
(a) Original	87.7	1.00	-	1.00	0.00
(b) μ	87.8	0.88	26.60	0.09	**664.31**
(c) $\mu + \sigma$	88.4	0.72	22.23	0.09	784.34
(d) $\mu + (\sigma$ x $\epsilon)$	**88.6**	**0.88**	**26.83**	**0.11**	726.94

Figure 16. **Qualitative results by Variance-aware method parameters.**

A.4.2 Comparison Results of images generated according to the number of iterations

The generated images depicted in Fig. 15(top) show a notable increase in the reflection of input text (class name) information as the number of training iterations in the model increases. This reduces the learning obstacles of the original image and allows for a more strongly represented representation of its properties. On the other hand, Fig. 15(bottom) reveals that achieved the highest level of performance between the 20th and 30th iterations. During the initial iteration, the standard deviation value is approximately 1.57, but it rapidly decreases as the model iterates, reaching 0.39. Therefore, it is essential to integrate relevant textual information under appropriate training iteration conditions to obtain optimal results.

A.4.3 Ablation studies by Variance-aware parameters

As shown by Table 16, Fig. 16, we compared the image quality effects through an ablation test of the variance-aware parameters elements Mu, sigma, and epsilon values. The SSIM, which considers visual elements such as image structure, contrast, and texture, was found to be most structurally similar when we reflected all parameters. The PSNR, which measures the difference between two images as the difference in pixel values, also shows the best score, which indicates good quality. VIF(Visual Information Fidelity), used to evaluate the quality of images with compression or loss, also shows the best score.

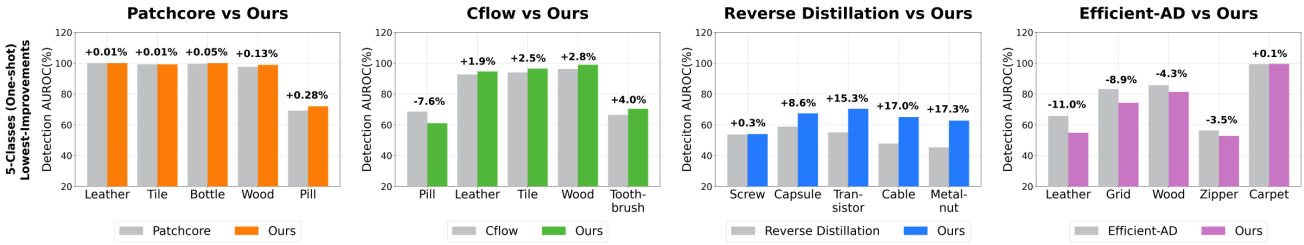

Figure 17. **Lowest-improving results for anomaly detection in MVTecAD dataset.** The results shows the average score for the lowest-improving five classes.

Table 17. **Performance improvement experiment.** Based on the lowest improvement results collected from Fig. 17, we attempt to improve the performance by changing the pre-trained CLIP model and augmentation strategy. (Augmentation strategies : Strategies-1 : RandomCrop, ColorJitter / Strategies-2 : RandomRotation, RandomAutocontrast)

Component	Leather(Texture)	Gain(%)	Pill(Object)	Gain(%)
Baseline	65.8(±0.0)	-	68.6(±0.0)	-
1) CLIP model				
ViT-B/16	54.8(±8.6)	(-11.0%)	61.0(±6.5)	(-7.6%)
ResNet50x64	**76.1(±0.1)**	**(+10.3%)**	**71.5(±0.0)**	**(+2.9%)**
2) Augmentation				
Strategies-1	65.8(±0.0)	(+0.0%)	68.6(±0.0)	(+0.0%)
Strategies-2	56.1(±0.0)	(-9.7%)	58.8(±0.0)	(-9.8%)

Table 18. **Comparison with Data Augmentation Strategies.** A one-shot scenario experiment was conducted on a toothbrush (Object type) and grid class (Texture type), and performance was compared to the baseline using Patchcore.

Component	Toothbrush(Object)	Gain(%)	Grid(Texture)	Gain(%)
Baseline	74.7	-	68.6	-
Strategies-1	76.9	(+2.2%)	61.8	(-6.8%)
Strategies-2	77.1	(+2.4%)	62.2	(-6.4%)
AutoAugmentation [6]	74.4	(-0.3%)	67.8	(-0.8%)
RandAugmentation [7]	76.7	(+2.0%)	70.3	(+1.7%)
Ours	**78.9**	**(+4.2%)**	**72.2**	**(+3.6%)**

A.5. Discussion

A.5.1 Lowest-improving results for anomaly detection in MVTecAD dataset.

Additionally, we find scenarios where performance was low and analyze how to improve them. In Fig. 17, the targets with the most significant decrease in performance were the Leather class in the 4th graph and the Pill class in the 2nd graph. As shown by Table 17, when we changed the CLIP model to ResNet50x64, the Leather class(texture type) was significantly improved to about 10.1%, and the variation also tended to change stably. However, when the augmentation strategy changed, it was ineffective in performance. Therefore, we analyzed the effectiveness of the augmentation strategy through additional experiments in Table 18.

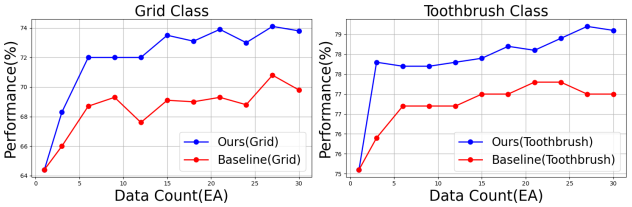

Figure 18. **Necessity of generating Non-defective data.** The addition of non-defective data(non-various images in Baseline) suffers from the rapid convergence of performance. However, through our generative model, a variety of non-defective data can be acquired to improve the performance, even with few real non-defective images.

A.5.2 Various augmentation strategies.

In Table 18, the possibility of improvement was analyzed by applying various augmentation strategies when generating non-defective images. Effective strategies differ depending on object type and texture type. For example, Strategies 1,2 presented were effective for object types such as Toothbrushes but not for texture types such as Grids. Therefore, we experimented with a random selection of Gaussian Blur, NoiseReduction, RandomRotation, RandomAdjustSharpness, RandomAutocontrast, and ColorJitter strategies that we empirically found to be effective. As a result, we confirmed the possibility of improving performance when specific strategies are used appropriately.

A.5.3 Design with connection to anomaly detection and segmentation.

Additional non-defective images do not necessarily increase the performance due to their duplicity, while the images generated by our approach show more effectiveness than the same number of real images. Our method is designed to generate non-defective data, preserving the possible variance of input data through text-based guidance. This approach is particularly relevant in industrial anomaly detection and segmentation, where outlier scores are based on the distance between test and non-defective images. The importance of non-defective generation is also validated through the experiments of Fig. 18.