Supplementary Material on

PNI : Industrial Anomaly Detection using Position and Neighborhood Information

S1. Implementation Details

The proposed PNI algorithm is implemented with Python 3.8 and PyTorch, version torch=1.12.1 and torchvision=0.13.1. The model is trained on NVIDIA TITAN RTX, A100, and T4 GPUs. We used ImageNet [8] pre-trained network from PyTorch/vision:v0.10.0. The WideResNet101-2 [31] network is used in our code by default, ResNext101 32x8d [27] and DenseNet201 [13] are used for ensemble results. In the implementation, the embedding coreset and the distribution coreset are stored in *faiss* [S2] framework to calculate the distance between a test feature and the coresets efficiently. We used *pytorch-lightning* [S1] framework to manage the training and evaluation process.

Figure S1 shows the more detailed process of training MLP for normal feature distribution given neighborhood information in Figure 2. With a pre-trained network ϕ, normal sample x_i is converted into feature map $\Phi_i \in \mathbb{R}^{C\times H\times W}$. With the spatial coordinate $\mathbf{x} = (h, w)$, we define neighborhood features $N_p(\mathbf{x})$ in equation (7) of the main paper. In Figure S1, we define $p = 9$ by default. All features in $N_p(\mathbf{x})$ are flattened and concatenated to 1-dimensional features to become an input of MLP. The MLP consists of $N_{\text{MLP}} = 10$ sequential linear layers with $c_{\text{MLP}} = 2048$ neurons by default. Batch normalization and ReLU functions are used between layers. The MLP outputs $|C_{\text{dist}}|$ nodes, which represents $p(c_{\text{dist}}|N_p(\mathbf{x}))$. The ground truth used for training is a one-hot vector, where the distribution coreset index closest to the true center feature vector is one, and the cross-entropy loss is calculated with the MLP output.

The implemented code is provided in a zip file as supplementary material. The model in the code trained on designated hyperparameters can achieve up to **99.56%** and **98.98%** AUROC scores in anomaly detection and localization for MVTec AD benchmark [1], which is the state-of-the-art performance. This model also achieves **96.05%** AUPRO score, which outperforms the second-best algorithm by 0.53%. In addition, the same model can achieve up to **97.8%** of anomaly localization AUROC for the BTAD [19], which is the highest performance compared to previous works.

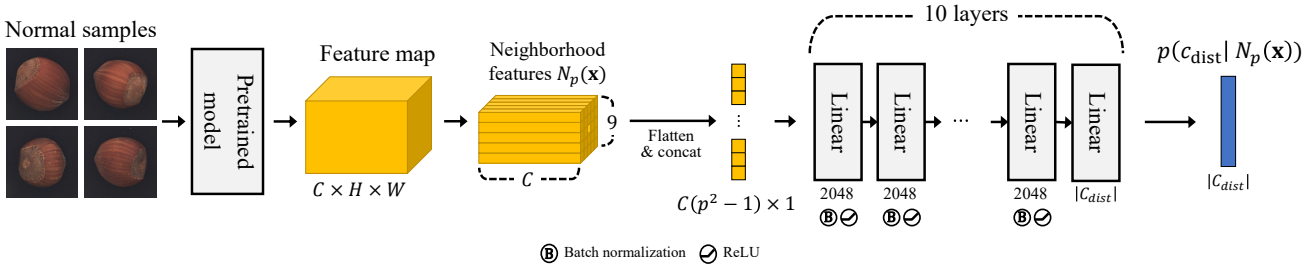

Figure S1: The detailed process of training MLP for normal feature distribution given neighborhood information. The numbers below the linear layer indicate the number of neurons. Each layer is annotated with the number of neurons, while the symbols for batch normalization and ReLU indicate the operations that follow each layer.

S2. Pixelwise Refinement Network

Detailed structure: Figure S2 shows the more detailed structure of the refinement network in Figure 4. $H \times W$ size image I and anomaly map $\hat{A}$ are transformed into $\frac{H}{4} \times \frac{W}{4}$ size features through a convolution layer and max pooling, respectively. These features are added element-wisely, forwarded to four dense blocks and three transition layers, and transformed into $2208 \times \frac{H}{32} \times \frac{W}{32}$ size features. The structure of dense blocks and transition layers are identical to that of DenseNet161 [13]. In the decoder, features are compressed to 768 channels by the first convolution layer and are upsampled to the original image size by five upsampling blocks. Each upsampling block consists of one bilinear upsampling interpolation and two following convolution layers. In an upsampling block, the spatial resolution of input features is expanded by factor 2, and the number of channels is halved. Thus, the output size of the fifth upsampling block becomes $24 \times H \times W$. The final convolution layer estimates $H \times W$ size refined anomaly map $\tilde{A}$.

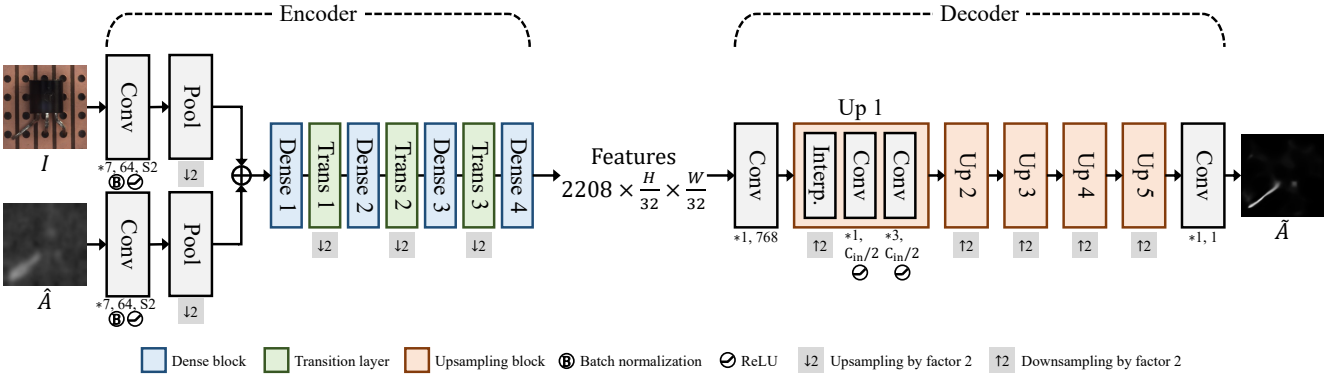

Figure S2: The architecture of the refinement network with detailed information on the convolution blocks. Each block is annotated with the kernel size, number of channels, and stride, while the symbols for batch normalization and ReLU indicate the operations that follow each convolution layer. Stride values are omitted when they equal 1. For example, '*7, 64, S2' indicates a kernel size of 7×7, 64 channels, and stride 2.

Defect images generated by manual drawing: Figure S3 shows generation examples generation. First, we make the defect patches in a hand-drawing manner as shown in Figure S3(a). Second, we generate artificial anomaly maps by combining patches at various sizes and spatial locations as shown in Figure S3(b). Finally, we obtain I from clean image I_{clean} and defect image I_{defect}.

$$I = (1 - A) \odot I_{\text{clean}} + A \odot I_{\text{defect}}, \tag{S1}$$

Figure S3(c) and (d) show examples of I_{clean} and I, respectively.

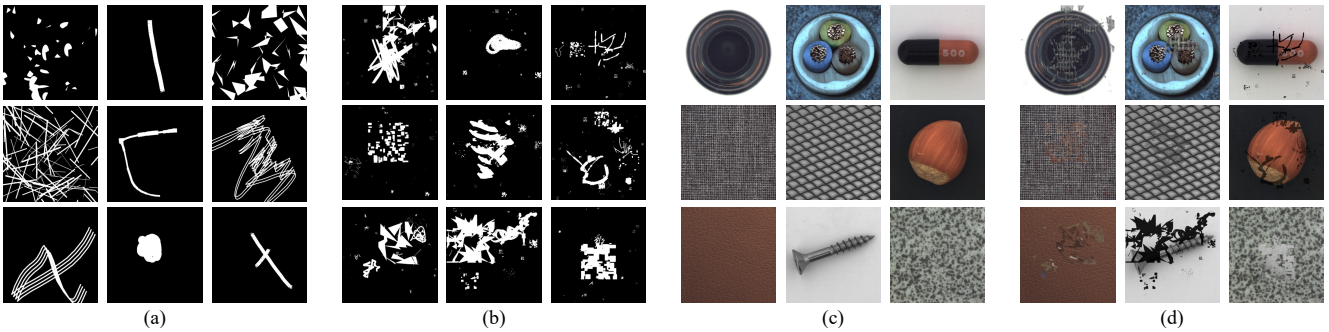

Figure S3: Manual defect image generation process according to manual drawing, (a) defect patches, (b) artificial anomaly maps, (c) I_{clean}, and (d) I_{defect}.

S3. Evaluation Results

In Table S2, S3, S4, and S5, we provide a more detailed quantitative comparison of anomaly detection and localization performance, which we do not cover in Tables 1 and 2 of the main paper due to space constraints. Some conventional algorithms provide multiple models with different hyperparameters. We specifically list in Table S1 which models we adopt for comparison from each algorithm. For the remaining algorithms, we adopt the representative models mentioned in their respective papers.

Table S1: Model selection of conventional algorithms for comparison.

Algorithm	Model
FCDD [S3]	Unsupervised FCDD
PaDiM [6]	PaDiM-WR50-Rd550
CutPaste [17]	CutPaste (3-way)
CutPaste (Ensemble) [17]	Ensemble
NSA [22]	NSA (logistic)
PatchCore [21]	PatchCore-25%
PatchCore (Ensemble) [21]	DenseN-201 & RNext-101 & WRN-101 (2+3), Imagesize 320
PEFM [25]	PEFM_{a}
Uninformed Students [2]	Multiscale
CFLOW-AD [9]	WideResNet-50

In Table S2, we compare the performance of the proposed PNI algorithm and conventional algorithms on the MVTec AD [1] dataset. To assess anomaly detection performance, image-level AUROC (I-AUROC) is used, while pixel-level AUROC (P-AUROC) and AUPRO are used for anomaly localization performance. Sub-total averages for object subcategories, texture subcategories, and overall averages are provided.

Table S2: Summary of anomaly detection and localization results on MVTec AD dataset for conventional algorithms. The proposed PNI is compared to recent algorithms in terms of I-AUROC, P-AUROC, and AUPRO. For each metric, sub-total averages are provided for both object and texture subcategories, additionally. For each metric, the best result is **boldfaced**, and the second best is underscored.

	I-AUROC			P-AUROC			AUPRO		
	Object	Texture	Average	Object	Texture	Average	Object	Texture	Average
FCDD [S3]	-	-	-	95	97	96	-	-	-
Patch SVDD [29]	90.8	94.5	92.1	96.7	93.7	95.7	-	-	-
SPADE [5]	-	-	85.5	97.6	92.9	96.0	93.4	88.4	91.7
PaDiM [6]	93.6	98.8	95.3	97.8	96.9	97.5	91.6	93.1	92.1
RIAD [33]	89.9	95.1	91.7	94.3	93.9	94.2	-	-	-
CutPaste [17]	94.4	97.0	95.2	95.8	96.3	96.0	-	-	-
CutPaste (ensemble) [17]	95.5	97.5	96.1	-	-	-	-	-	-
DRÆM [32]	97.4	99.1	98.0	97.0	97.9	97.3	-	-	-
FastFlow [30]	99.1	**99.9**	99.4	98.6	98.1	98.5	-	-	-
SOMAD [18]	97.7	98.4	97.9	98.1	97.1	97.8	94.1	91.6	93.3
InTra [20]	93.1	98.9	95.0	96.9	96.1	96.6	-	-	-
MB-PFM [26]	-	99.4	97.5	97.0	97.8	97.3	92.3	94.6	93.0
NSA [22]	96.5	98.6	97.2	96.0	96.8	96.3	90.4	92.2	91.0
IKD [3]	-	-	-	98.3	96.8	97.8	93.3	91.1	92.5
PatchCore [21]	99.2	99.0	99.1	98.4	97.5	98.1	93.3	93.6	93.4
PatchCore (ensemble) [21]	-	-	99.6	-	-	98.2	-	-	94.9
Reverse Distillation [7]	98.0	99.5	98.5	97.9	97.7	97.8	93.4	95.0	93.9
Tsai *et al*. [24]	98.4	97.7	98.1	98.4	97.6	98.1	95.7	95.0	95.5
PEFM [25]	-	-	-	98.37	98.17	98.30	95.30	95.95	95.52
CDO [4]	-	-	-	98.36	97.94	98.22	94.57	94.90	94.68
Uninformed Students [2]	-	-	-	-	-	-	90.8	92.7	91.4
CFLOW-AD [9]	97.66	99.47	98.26	98.69	98.51	98.62	93.58	**96.65**	94.60
PNI	99.55	99.59	99.56	99.12	98.72	98.98	96.34	95.47	96.05
PNI (Ensemble)	**99.64**	99.59	**99.63**	**99.14**	**98.90**	**99.06**	**96.83**	96.00	**96.55**

Unlike the main paper, we also compare models that use multiple networks alongside other models, and we indicate the best and second best performances with **boldface** and underscore, respectively. The proposed PNI outperforms all other conventional algorithms in 7 out of 9 metrics, excluding I-AUROC texture and AUPRO texture. Furthermore, PNI (Ensemble) improves PNI in all terms, and notably, it demonstrates performance exceeding 99% in overall P-AUROC for the first time. In Table S3, S4, and S5, we provide the I-AUROC, P-AUROC, and AUPRO performance of individual subcategories, respectively. These tables include detailed results of the algorithms that are not covered in the main paper. Note that in terms of I-AUROC, PaDiM [6] does not provide performance for each subcategory. Also, MB-PFM [26] is missing the result for the capsule subcategory.

As shown in Table S3, the PNI algorithm also demonstrates the best anomaly detection performance in the subcategories. For example, out of 15 subcategories, PNI and PNI (Ensemble) show 100% detection results in 7 and 8 subcategories, respectively, which surpasses the 5 subcategories for MB-PFM [26] and 4 subcategories for Reverse Distillation [7]. Additionally, PNI outperforms the anomaly detection performance of conventional algorithms in 11 subcategories.

Table S3: Comparison of anomaly detection performance between the proposed PNI algorithm and conventional algorithms on the MVTec AD dataset using I-AUROC. Performance for each subcategory is also provided. For each subcategory, the best result is **boldfaced**.

		Patch SVDD	PaDiM	DRÆM	SOMAD	MB-PFM	NSA	Reverse Distillation	CutPaste (Ensemble)	**PNI**	**PNI (Ensemble)**
Object	Bottle	98.6	-	99.2	**100**	**100**	97.7	**100**	98.2	**100**	**100**
	Cable	90.3	-	91.8	98.8	98.8	94.5	95.0	81.2	99.76	**99.91**
	Capsule	76.7	-	98.5	93.8	-	95.2	96.3	98.2	**99.72**	**99.72**
	Hazelnut	92.0	-	**100**	**100**	**100**	94.7	99.9	98.3	**100**	**100**
	Metal nut	94.0	-	98.7	99.7	**100**	98.7	**100**	99.9	**100**	**100**
	Pill	86.1	-	98.9	98.6	96.5	**99.2**	96.6	94.9	96.89	97.79
	Screw	81.3	-	93.9	95.5	91.8	90.2	97.0	88.7	**99.51**	99.10
	Toothbrush	**100**	-	**100**	98.6	88.6	**100**	99.5	99.4	99.72	**100**
	Transistor	91.5	-	93.1	94.5	97.8	95.1	96.7	96.1	**100**	**100**
	Zipper	97.9	-	**100**	97.7	97.4	99.8	98.5	99.9	99.87	98.89
	Average	90.8	93.6	97.4	97.7	-	96.5	98.0	95.5	99.55	**99.64**
Texture	Carpet	92.9	-	97.0	**100**	**100**	95.6	98.9	93.9	**100**	**100**
	Grid	94.6	-	99.9	93.9	98.0	99.9	**100**	**100**	98.41	98.50
	Leather	90.9	-	**100**	**100**	**100**	99.9	**100**	**100**	**100**	**100**
	Tile	97.8	-	99.6	98.7	99.6	**100**	99.3	94.6	**100**	**100**
	Wood	96.5	-	99.1	99.2	99.5	97.5	99.2	99.1	**99.56**	99.47
	Average	94.5	98.8	99.1	98.4	99.4	98.6	99.5	97.5	**99.59**	**99.59**
Average		92.1	95.3	98.0	97.9	97.5	97.2	98.5	96.1	99.56	**99.63**

In Tables S4 and S5, which assess anomaly localization performance using P-AUROC and AUPRO, respectively, the proposed PNI demonstrates outstanding performance in multiple subcategories. In terms of P-AUROC, PNI shows the best or second-best results in 9 out of 15 subcategories and exhibits performance above 99% in 11 subcategories. This surpasses the competitor ones [4,7,22,25,32], which only show performance exceeding 99% in 3 to 6 subcategories. Furthermore, PNI (Ensemble) presents the best or second-best results in 11 out of 15 subcategories. In AUPRO, PNI (Ensemble) ranks within the top two in 8 subcategories, while showing performance above 95% in 12 subcategories. This also surpasses the other algorithms [2,4,7,9,24,25], which record performance above 95% in 7 to 11 subcategories.

Table S6 presents the anomaly detection and localization performance of the proposed PNI algorithm on the VisA dataset using I-AUROC and P-AUROC. Scores are provided for each subcategory, sub-total average of each subcategory type, and overall average

Table S4: Comparison of anomaly localization performance between the proposed PNI algorithm and conventional algorithms on the MVTec AD dataset using P-AUROC. Performance for each subcategory is also provided. For each subcategory, the best result is **boldfaced**, and the second best is underscored.

		FCDD	Patch SVDD	SPADE	PaDiM	DRÆM	SOMAD	MB-PFM	NSA	IKD	Reverse Distillation	PEFM	CDO	**PNI**	**PNI (Ensemble)**
Object	Bottle	96	98.1	98.4	98.3	99.1	98.3	98.4	98.3	98.99	98.7	98.51	**99.30**	98.87	99.03
	Cable	93	96.8	97.2	96.7	94.7	98.2	96.7	96.0	98.03	97.4	98.31	97.60	99.10	**99.16**
	Capsule	95	95.8	99.0	98.5	94.3	98.7	98.3	97.6	98.55	98.7	98.51	98.64	99.34	**99.38**
	Hazelnut	97	97.5	99.0	98.2	**99.7**	98.4	99.1	97.6	98.71	98.9	99.17	99.24	99.37	99.40
	Metal nut	98	98.0	98.1	97.2	**99.5**	98.0	97.2	98.4	98.38	97.3	96.98	98.54	99.29	99.34
	Pill	97	95.1	96.5	95.7	97.6	98.0	97.2	98.5	98.79	98.2	97.04	98.94	**99.03**	98.99
	Screw	93	95.7	98.9	98.5	97.6	99.1	98.7	96.5	98.63	99.6	99.01	99.01	99.60	**99.68**
	Toothbrush	95	98.1	97.9	98.8	98.1	98.5	98.6	94.9	98.58	99.1	**99.18**	98.86	99.09	99.11
	Transistor	90	97.0	94.1	97.5	90.9	95.3	87.8	88.0	97.13	92.5	**98.39**	95.30	98.04	97.74
	Zipper	98	95.1	96.5	98.5	98.8	98.7	98.2	94.2	97.56	98.2	98.61	98.21	99.43	**99.56**
	Average	95	96.7	97.6	97.8	97.0	98.1	97.0	96.0	98.34	97.9	98.37	98.36	99.12	**99.14**
Texture	Carpet	99	92.6	97.5	99.1	95.5	98.9	99.2	95.5	98.71	98.9	99.15	99.08	99.40	**99.46**
	Grid	95	96.2	93.7	97.3	**99.7**	98.4	98.8	99.2	97.04	99.3	99.23	98.40	99.20	99.20
	Leather	99	97.4	97.6	99.2	98.6	99.1	99.4	99.5	98.53	99.4	99.42	99.17	99.56	**99.59**
	Tile	98	91.4	87.4	94.1	99.2	94.8	96.2	**99.3**	95.68	95.6	96.55	97.20	98.40	98.69
	Wood	94	90.8	88.5	94.9	96.4	94.4	95.6	90.7	93.88	95.3	96.49	95.85	97.04	**97.55**
	Average	97	93.7	92.9	96.9	97.9	97.1	97.8	96.8	96.77	97.7	98.17	97.94	98.72	**98.90**
Average		96	95.7	96.0	97.5	97.3	97.8	97.3	96.3	97.81	97.8	98.30	98.22	98.98	**99.06**

Table S5: Comparison of anomaly localization performance between the proposed PNI algorithm and conventional algorithms on the MVTec AD dataset using AUPRO. Performance for each subcategory is also provided. For each subcategory, the best result is **boldfaced**, and the second best is underscored.

		SPADE	PaDiM	SOMAD	MB-PFM	NSA	IKD	PatchCore	Reverse Distillation	Tsai *et al.*	PEFM	CDO	Uninformed Students	CFLOW -AD	**PNI**	**PNI (Ensemble)**
Object	Bottle	95.5	94.8	94.7	95.4	92.9	96.08	96.2	96.6	95.3	95.92	97.17	93.1	96.80	95.95	**96.84**
	Cable	90.9	88.8	93.4	94.2	89.9	94.21	92.5	91.0	96.7	97.73	94.17	81.8	93.53	98.93	**99.23**
	Capsule	93.7	93.5	93.4	91.7	91.4	90.62	95.5	95.8	**97.8**	92.11	92.97	96.8	93.40	95.63	96.12
	Hazelnut	95.4	92.6	95.1	96.7	93.6	95.97	93.8	95.5	97.8	**97.99**	97.39	96.5	96.68	96.93	97.35
	Metal nut	94.4	85.6	93.6	94.6	94.6	94.69	91.4	92.3	88.8	93.88	95.74	94.2	91.65	95.89	**96.88**
	Pill	94.6	92.7	96.5	96.1	96.0	96.09	93.2	96.4	96.1	96.18	96.59	96.1	95.39	96.68	**97.00**
	Screw	96.0	94.4	96.0	93.4	90.1	92.95	97.9	98.2	**98.3**	95.73	94.33	94.2	95.30	97.17	97.88
	Toothbrush	93.5	93.1	90.7	90.7	90.7	87.01	91.5	94.5	94.4	**96.21**	90.50	93.3	95.06	92.68	93.76
	Transistor	87.4	84.5	91.6	74.9	75.3	93.78	83.7	78.0	95.0	90.84	92.56	66.6	81.40	**96.24**	95.35
	Zipper	92.6	95.9	95.9	94.8	89.2	91.55	97.1	95.4	97.0	96.45	94.28	95.1	96.60	97.28	**97.86**
	Average	93.4	91.6	94.1	92.3	90.4	93.30	93.3	93.4	95.7	95.30	94.57	90.8	93.58	96.34	**96.83**
Texture	Carpet	94.7	96.2	95.5	96.9	85.0	94.49	96.6	97.0	92.7	96.75	96.77	87.9	**97.70**	97.55	97.67
	Grid	86.7	94.6	95.3	96.0	96.8	87.73	96.0	97.6	**97.9**	97.21	96.02	95.2	96.08	94.26	94.29
	Leather	97.2	97.8	97.7	98.8	98.7	97.64	98.9	99.1	99.2	98.91	98.34	94.5	**99.35**	98.27	98.58
	Tile	75.9	86.0	81.3	88.7	95.3	86.35	87.3	90.6	88.8	91.10	90.51	94.6	94.34	94.74	**95.66**
	Wood	87.4	91.1	88.2	92.6	85.3	89.06	89.4	90.9	**96.2**	95.77	92.87	91.1	95.79	92.51	93.82
	Average	88.4	93.1	99.4	94.6	92.2	91.05	93.6	95.0	95.0	95.95	94.90	92.7	**96.65**	95.47	96.00
Average		91.7	92.1	97.5	93.0	91.0	92.55	93.4	93.9	95.5	95.52	94.68	91.4	94.60	96.05	**96.55**

Table S6: I-AUROC and P-AUROC scores of the proposed PNI algorithm on VisA [35] are presented. Scores are provided for each subcategory, sub-total average of each subcategory type, and overall average.

	Single instance					Multiple instances					Complex structure					Average
	Cashew	Chewing gum	Fryum	Pipe fyrum	Average	Macaroni1	Macaroni2	Capsules	Candle	Average	PCB1	PCB2	PCB3	PCB4	Average	
I-AUROC	99.04	99.06	98.94	99.74	99.20	94.66	74.34	83.27	99.33	87.90	98.84	97.52	97.95	99.84	98.54	95.21
P-AUROC	99.18	99.01	94.68	99.37	98.06	99.67	98.56	99.10	99.54	99.22	99.80	98.92	99.00	98.35	99.02	98.77

S4. Ablation Study

We conduct detailed ablation studies on the components of the proposed PNI algorithm, which is covered in Table 3 of the main paper. Table S7 shows the specific settings of each model, named setting A, B, ..., and L. These settings cover various ablation studies on the three main components of the proposed PNI algorithm, which are neighborhood information, position information, and pixelwise refinement, as well as coreset subsampling ratios, defect image creation methods, and a different loss setting. For example, setting A is the baseline of the ablation study, which does not include neighborhood, position information, or pixelwise refinement. Table S8 summarizes the performance of the settings on the MVTec AD dataset, while Tables S9, S10, and S11 provide more detailed information on I-AUROC, P-AUROC, and AUPRO for each subcategory, respectively.

Table S7: Various settings for the ablation study of the proposed PNI algorithm, in which we break down the PNI into detailed components and define settings A, B, ..., and L, using only specific combinations of these components.

	Neighborhood	Position	Refinement	Subsampling ratio	Defect image creation method				Loss ℓ_{grad}	Ensemble
					CutPaste	CutPaste (scar)	DRÆM	Manual		
Setting A (baseline)	-	-	-	0.01	-	-	-	-	-	-
Setting B	✓	-	-	0.01	-	-	-	-	-	-
Setting C	✓	✓	-	0.0025	-	-	-	-	-	-
Setting D	✓	✓	-	0.005	-	-	-	-	-	-
Setting E	✓	✓	-	0.01	-	-	-	-	-	-
Setting F	✓	✓	-	0.02	-	-	-	-	-	-
Setting G	✓	✓	✓	0.01	✓	-	-	-	✓	-
Setting H	✓	✓	✓	0.01	-	✓	-	-	✓	-
Setting I	✓	✓	✓	0.01	✓	✓	-	-	✓	-
Setting J	✓	✓	✓	0.01	-	-	✓	-	✓	-
Setting K	✓	✓	✓	0.01	-	-	-	✓	✓	-
Setting L	✓	✓	✓	0.01	✓	✓	✓	✓	-	-
PNI	✓	✓	✓	0.01	✓	✓	✓	✓	✓	-
PNI (ensemble)	✓	✓	✓	0.01	✓	✓	✓	✓	✓	✓

Three main components: By comparing the settings A, B, E, and PNI in Tables S9, S10, and S11, we can observe the effects of the three components of the PNI algorithm, which are neighborhood information, position information, and pixelwise refinement. These results provide a more detailed version of Table 3 in the main paper, and the following observations can be made.

- The setting A without the three components is identical to PatchCore and performs similarly.
- The setting B, E, and PNI outperform setting A in terms of I-AUROC, P-AUROC, and AUPRO since setting A deals with normal features unconditionally.
- As shown in the comparison between settings A and B, the use of neighborhood information significantly improves performance in all metrics, particularly in texture subcategories where greater improvements are observed. For example, in the carpet subcategory, AUPRO is improved by 22.88%, increasing from 71.94% to 94.82%.
- As shown in the comparison between settings B and E (which is consistent with intuition mentioned in the main paper), position information is effective for object subcategories. For example, in the transistor subcategory, improvements of 2.19% and 3.1% are observed in P-AUROC and AUPRO, respectively.
- Pixelwise refinement is complementary to the position information and is more effective in texture subcategories. For example, when comparing the settings E and PNI, improvements of 0.26%, 0.85%, and 1.71% in I-AUROC, P-AUROC, and AUPRO, respectively, are observed for the wood subcategory.

Coreset subsampling ratio: We compare the effects of coreset sampling ratios of 0.25%, 0.5%, 1%, and 2% in settings C, D, E, and F. Generally, increasing the sampling ratio tends to improve anomaly detection and localization performance. However, the gain from increasing the sampling ratio converges. For example, when comparing settings E and F, setting F shows better anomaly localization performance but worse anomaly detection performance. In the subcategory level, F performs worse than E in the pill, screw, zipper, grid, and tile subcategories in terms of I-AUROC. On the other hand, increasing the sampling ratio significantly increases the runtime of the algorithm. Setting F requires approximately twice the time for inference compared to setting E. Considering both performance and inference time, we adopt the optimal coreset sampling ratio to 1%.

Table S8: Summary of anomaly detection and localization results on MVTec AD dataset for ablation studies. For each metric, the best result is **boldfaced**, and the second best is underscored.

	I-AUROC			P-AUROC			AUPRO		
	Object	Texture	Average	Object	Texture	Average	Object	Texture	Average
Setting A (baseline)	99.01	98.75	98.92	98.70	97.15	98.18	92.30	85.09	89.90
Setting B	99.38	99.55	99.44	98.79	98.29	98.62	94.98	93.86	94.61
Setting C	97.86	99.44	98.39	98.68	98.24	98.53	92.09	93.64	92.61
Setting D	99.39	99.35	99.38	98.95	98.30	98.73	94.52	93.79	94.28
Setting E	99.46	99.46	99.46	99.03	98.29	98.79	95.27	93.79	94.78
Setting F	99.41	99.45	99.42	99.05	98.30	98.80	95.34	93.82	94.83
Setting G	99.18	99.40	99.26	98.99	98.34	98.77	95.69	94.18	95.19
Setting H	99.27	99.50	99.35	99.04	98.33	98.80	96.04	94.09	95.39
Setting I	99.46	99.40	99.44	99.07	98.37	98.84	95.86	94.14	95.29
Setting J	99.57	**99.59**	99.58	99.05	98.69	98.93	96.16	95.65	95.99
Setting K	99.48	99.62	99.53	99.12	98.56	98.93	95.69	94.74	95.38
Setting L	99.44	99.57	99.48	99.07	98.68	98.94	96.33	95.42	96.02
PNI	99.55	**99.59**	99.56	99.12	98.72	98.98	96.34	95.47	96.05
PNI (Ensemble)	**99.64**	**99.59**	**99.63**	**99.14**	**98.90**	**99.06**	**96.83**	**96.00**	**96.55**

Table S9: Comparison of anomaly detection performance between ablation settings and the proposed PNI algorithm on the MVTec AD dataset using I-AUROC. Performance for each subcategory is also provided. For each subcategory, the best result is **boldfaced**.

		Setting												Proposed	
		A	B	C	D	E	F	G	H	I	J	K	L	**PNI**	**PNI (Ensemble)**
Object	Bottle	**100**	**100**	**100**	**100**	**100**	**100**	**100**	99.92	**100**	**100**	**100**	**100**	**100**	**100**
	Cable	99.63	99.04	99.63	99.79	99.42	99.48	99.46	99.39	99.40	99.44	99.68	99.66	99.76	**99.91**
	Capsule	98.92	99.36	99.24	99.36	99.44	99.60	99.44	**99.76**	99.44	99.56	99.48	99.60	99.72	99.72
	Hazelnut	**100**	**100**	99.11	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**
	Metal nut	**100**	**100**	99.90	99.51	**100**	**100**	**100**	**100**	**100**	**100**	99.95	**100**	**100**	**100**
	Pill	95.23	96.86	97.23	97.25	96.97	96.56	96.89	96.67	96.97	97.49	96.54	97.30	96.89	**97.79**
	Screw	96.37	99.55	97.54	99.34	**99.57**	99.28	97.66	99.00	99.55	99.49	99.30	99.55	99.51	99.10
	Toothbrush	**100**	99.17	86.39	98.89	99.44	99.44	99.44	99.44	99.44	**100**	**100**	98.61	99.72	**100**
	Transistor	**100**	**100**	**100**	**100**	**100**	**100**	99.96	**100**	**100**	99.96	**100**	**100**	**100**	**100**
	Zipper	**99.97**	99.82	99.58	99.74	99.76	99.74	98.98	98.56	99.82	99.76	99.84	99.63	99.87	99.89
	Average	99.01	99.38	97.86	99.39	99.46	99.41	99.18	99.27	99.46	99.57	99.48	99.44	99.55	**99.64**
Texture	Carpet	96.99	99.60	99.80	99.68	99.80	99.84	99.68	99.80	99.68	**100**	**100**	**100**	**100**	**100**
	Grid	98.41	**98.75**	98.33	98.33	98.41	98.33	97.91	98.41	98.41	98.41	98.58	98.41	98.41	98.50
	Leather	**100**	99.83	99.76	99.56	99.83	99.90	**100**	99.93	99.66	99.97	**100**	**100**	**100**	**100**
	Tile	98.70	**100**	99.93	99.89	99.96	99.89	99.96	99.82	99.96	**100**	99.96	99.89	**100**	**100**
	Wood	**99.65**	99.56	99.39	99.30	99.30	99.30	99.47	99.56	99.30	99.56	99.56	99.56	99.56	99.47
	Average	98.75	99.55	99.44	99.35	99.46	99.45	99.40	99.50	99.40	99.59	**99.62**	99.57	99.59	99.59
Average		98.92	99.44	98.39	99.38	99.46	99.42	99.26	99.35	99.44	99.58	99.53	99.48	99.56	**99.63**

Defect image creation method: We compare the effects of different defect image creation methods for training the pixelwise refinement network in settings G, H, I, J, K, and PNI. In settings G, H, J, and K, we train the refinement network using only the defect images created by the CutPaste [17], CutPaste (scar) [17], DRÆM [32], and manual drawing methods, respectively. Examples of defect images created by each method are shown in Figure 3 of the main paper. In setting I, we use a combination of the CutPaste and CutPaste (scar) methods to create the defect images, which is the method adopted in CutPaste (3-way) [17].

As shown in settings G and H, the methods using relatively simple defect types, from CutPaste and CutPaste (scar), are not suitable for training the refinement network. Even when compared to the results of setting E without refinement, there is a little decrease in performance in settings G and H. On the other hand, in setting I where two defect creation methods are combined, an improvement in anomaly localization performance is observed. For example, setting I shows a 0.51% improvement in AUPRO compared to setting E. These results imply that using a combination of diverse defect patterns for training can enhance the effectiveness of pixelwise refinement.

In settings J and K, which generate more complex defect images based on Perlin noise [S4] or manual drawing, significant performance improvement is observed especially in anomaly localization performance in texture subcategories. Finally, in the PNI algorithm that combines all 4 defect image creation methods mentioned earlier, the training of the pixelwise refinement network works most effectively, and once again, it demonstrates the effectiveness of the approach of PNI combining multiple synthetic defect data.

Position Information: While using position information is beneficial to most aligned object classes, it shows little improvement in some classes such as screw, which are not aligned. (Compare settings B and E in Tables S9, S10, and S11.) In practical industrial environments, however, the alignment of rigid objects can be performed during the preprocessing stage without challenge, which makes the proposed PNI work effectively for certain objects.

Table S10: Comparison of anomaly localization performance between ablation settings and the proposed PNI algorithm on the MVTec AD dataset using P-AUROC. Performance for each subcategory is also provided. For each subcategory, the best result is **boldfaced**, and the second best is underscored.

		Setting												Proposed	
		A	B	C	D	E	F	G	H	I	J	K	L	**PNI**	**PNI (Ensemble)**
Object	Bottle	98.54	98.90	98.83	98.88	98.90	98.90	98.74	98.79	98.79	98.86	98.93	98.81	98.87	**99.03**
	Cable	98.68	98.75	98.90	99.01	99.07	99.11	99.06	99.07	99.07	99.06	**99.17**	99.09	99.10	99.16
	Capsule	99.06	99.26	99.20	99.25	99.26	99.25	99.29	99.32	99.31	99.31	99.36	99.32	99.34	**99.38**
	Hazelnut	98.82	99.13	98.86	99.10	99.15	99.14	99.09	99.10	99.36	99.17	99.18	99.20	99.37	**99.40**
	Metal nut	99.16	99.37	99.09	99.14	99.31	99.32	99.26	99.29	99.29	99.27	99.27	99.27	99.29	**99.34**
	Pill	98.81	98.89	98.55	98.80	98.89	98.90	98.99	99.02	99.02	99.02	**99.06**	99.05	99.03	98.99
	Screw	99.04	99.53	99.06	99.44	99.53	99.57	99.54	99.59	99.57	99.60	99.35	99.59	99.60	**99.68**
	Toothbrush	98.79	99.08	97.98	99.01	99.08	99.10	98.99	99.01	98.99	99.03	99.02	**99.13**	99.09	99.11
	Transistor	97.41	95.67	97.08	97.54	97.86	97.87	97.99	97.88	97.99	97.86	**98.55**	97.90	98.04	97.74
	Zipper	98.69	99.32	99.29	99.30	99.31	99.31	98.95	99.33	99.33	99.35	99.31	99.37	99.43	**99.56**
	Average	98.70	98.79	98.68	98.95	99.03	99.05	98.99	99.04	99.07	99.05	99.12	99.07	99.12	**99.14**
Texture	Carpet	98.42	99.22	99.21	99.22	99.25	99.24	99.29	99.27	99.33	99.45	99.41	99.42	99.40	**99.46**
	Grid	97.50	98.80	98.68	98.79	98.77	98.77	98.94	98.99	98.97	99.19	99.12	**99.20**	**99.20**	**99.20**
	Leather	99.18	99.49	99.48	99.47	99.48	99.48	99.55	99.49	99.53	99.53	99.54	99.55	99.56	**99.59**
	Tile	96.53	97.66	97.73	97.78	97.78	97.81	97.63	97.64	97.74	98.37	98.20	98.21	98.40	**98.69**
	Wood	94.12	96.26	96.08	96.22	96.19	96.22	96.29	96.26	96.29	96.91	96.52	97.02	97.04	**97.55**
	Average	97.15	98.29	98.24	98.30	98.29	98.30	98.34	98.33	98.37	98.69	98.56	98.68	98.72	**98.90**
Average		98.18	98.62	98.53	98.73	98.79	98.80	98.77	98.80	98.84	98.93	98.93	98.94	98.98	**99.06**

Loss ℓ_{grad}: To show the efficacy of using ℓ_{grad} in (11) of the main paper, we compare the setting L and PNI. ℓ_{grad} makes the training of the refinement network more focused on a near edge region of defect, and it is effective for anomaly detection results. For example, in 13 of 15 subcategories, PNI keeps or improves I-AUROC performance compared to setting L.

Table S11: Comparison of anomaly localization performance between ablation settings and the proposed PNI algorithm on the MVTec AD dataset using AUPRO. Performance for each subcategory is also provided. For each subcategory, the best result is **boldfaced**, and the second best is underscored.

		Setting												Proposed	
		A	B	C	D	E	F	G	H	I	J	K	L	**PNI**	**PNI (Ensemble)**
Object	Bottle	90.14	95.32	95.09	95.34	95.29	95.52	95.34	95.80	95.45	95.77	94.32	96.17	95.95	**96.84**
	Cable	96.49	98.27	97.88	98.53	98.77	99.06	98.70	98.72	98.70	98.76	98.81	98.87	98.93	**99.23**
	Capsule	92.92	95.15	93.31	93.86	95.10	94.42	95.17	95.47	95.30	95.44	95.02	95.82	95.63	**96.12**
	Hazelnut	83.41	90.03	82.89	87.97	90.06	89.82	96.57	96.54	96.40	96.89	96.02	96.90	96.93	**97.35**
	Metal nut	93.20	95.32	93.59	92.83	94.88	95.22	95.33	95.23	95.30	95.78	94.95	95.80	95.89	**96.88**
	Pill	93.92	96.76	96.22	96.44	96.75	96.78	96.44	96.54	96.53	96.72	96.03	96.74	96.68	**97.00**
	Screw	94.05	97.09	93.75	96.36	97.05	97.28	96.88	97.14	97.06	97.18	96.32	97.18	97.17	**97.88**
	Toothbrush	91.03	91.90	76.14	91.27	91.88	92.28	91.54	91.95	90.83	91.95	92.54	92.54	92.68	**93.76**
	Transistor	94.04	92.96	95.30	95.70	96.06	95.97	96.13	96.09	96.15	96.06	96.20	96.15	**96.24**	95.35
	Zipper	93.82	96.96	96.75	96.90	96.90	97.02	94.81	96.92	96.86	97.02	96.70	97.11	97.28	**97.86**
	Average	92.30	94.98	92.09	94.52	95.27	95.34	95.69	96.04	95.86	96.16	95.69	96.33	96.34	**96.83**
Texture	Carpet	71.94	94.82	95.20	95.02	94.89	95.23	96.91	96.61	96.67	**97.70**	97.46	97.69	97.55	97.67
	Grid	89.21	93.58	92.49	93.28	93.29	93.12	92.91	92.83	93.18	94.11	93.07	**94.52**	94.26	94.29
	Leather	95.60	97.83	97.79	97.71	97.77	97.78	98.17	97.90	97.66	98.15	98.13	98.34	98.27	**98.58**
	Tile	86.85	92.10	92.08	92.20	92.21	92.18	92.08	92.13	92.41	**95.71**	93.65	94.04	94.74	95.66
	Wood	81.84	90.98	90.64	90.73	90.80	90.82	90.85	90.96	90.80	92.59	91.41	92.49	92.51	**93.82**
	Average	85.09	93.86	93.64	93.79	93.79	93.82	94.18	94.09	94.14	95.65	94.74	95.42	95.47	**96.00**
Average		89.90	94.61	92.61	94.28	94.78	94.83	95.19	95.39	95.29	95.99	95.38	96.02	96.05	**96.55**

Precision-recall curve: Similar to Figure 5 of the main paper, we show the pixel-level precision-recall curves and F1-max scores for each ablation setting across all subcategories of the MVTec AD in Figure S4. Again, each component of the algorithm improves anomaly localization performance in the precision-recall curves as well.

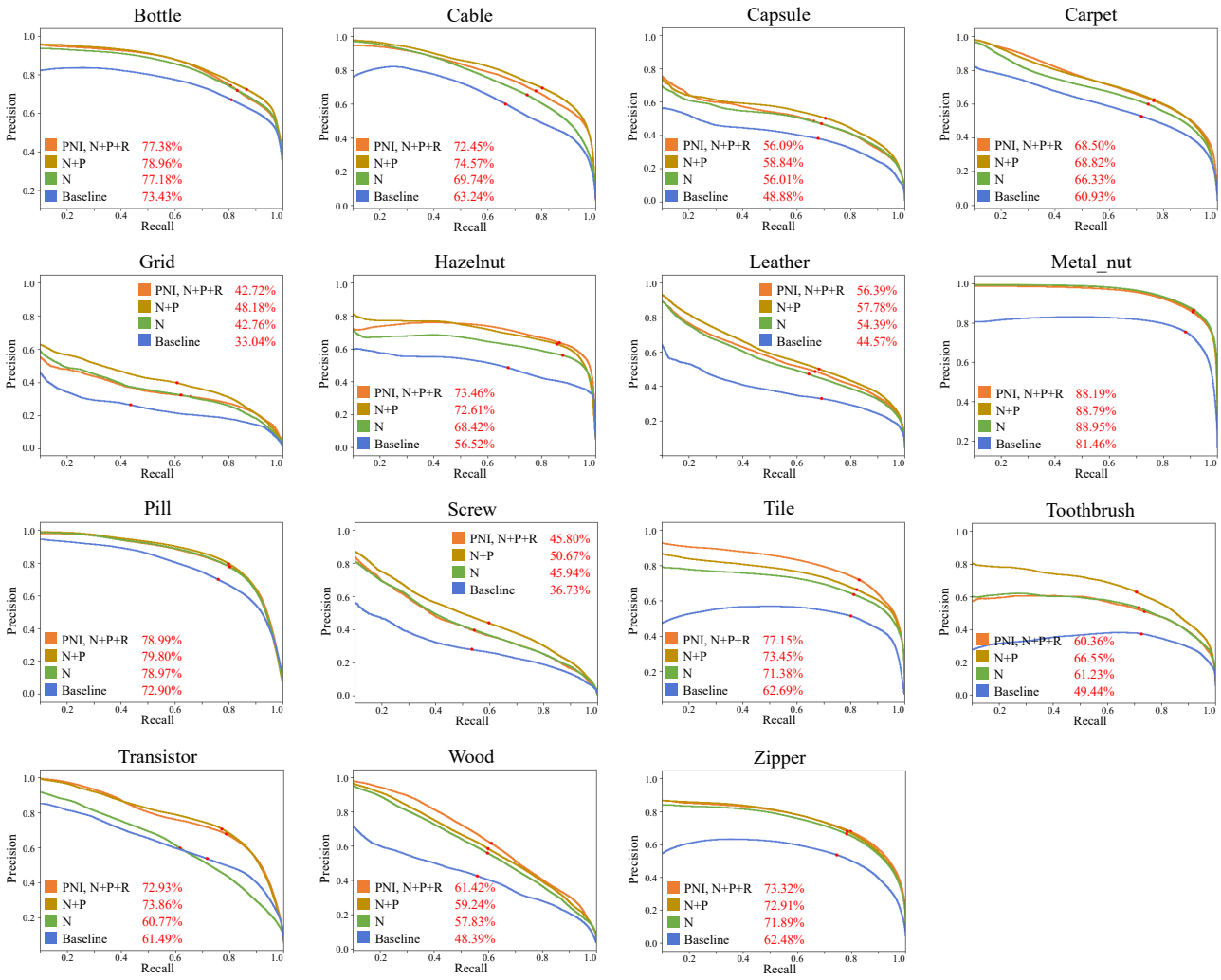

Figure S4: Precision-recall curve at the pixel-level for 15 subcategories of MVTec AD. The proposed PNI algorithm and three ablation settings Setting A, B, and E in Table 3 of the main paper, are compared. The F1-max scores for each setting are indicated on the right side of the legend.

S5. Qualitative Results

S5.1. Misclassified Samples

The proposed PNI with ensemble method achieves 99.66% anomaly detection AUROC (I-AUROC) on MVTec AD benchmark as shown in Table S2. We examine all misclassified samples on the dataset to analyze the limitation of our model. We compute false-positive and false-negative samples with the threshold optimizing F1 scores of anomaly detection. With these per-category thresholds, total of 7 false-positive errors and 12 false-negative errors are found from the test dataset from 467 normal test images and 1258 defective test images, which are shown in Figure S5 and Figure S6, respectively. The corresponding false negative rate (FNR) and false positive rate (FPR) are 0.95% and 1.50%, respectively.

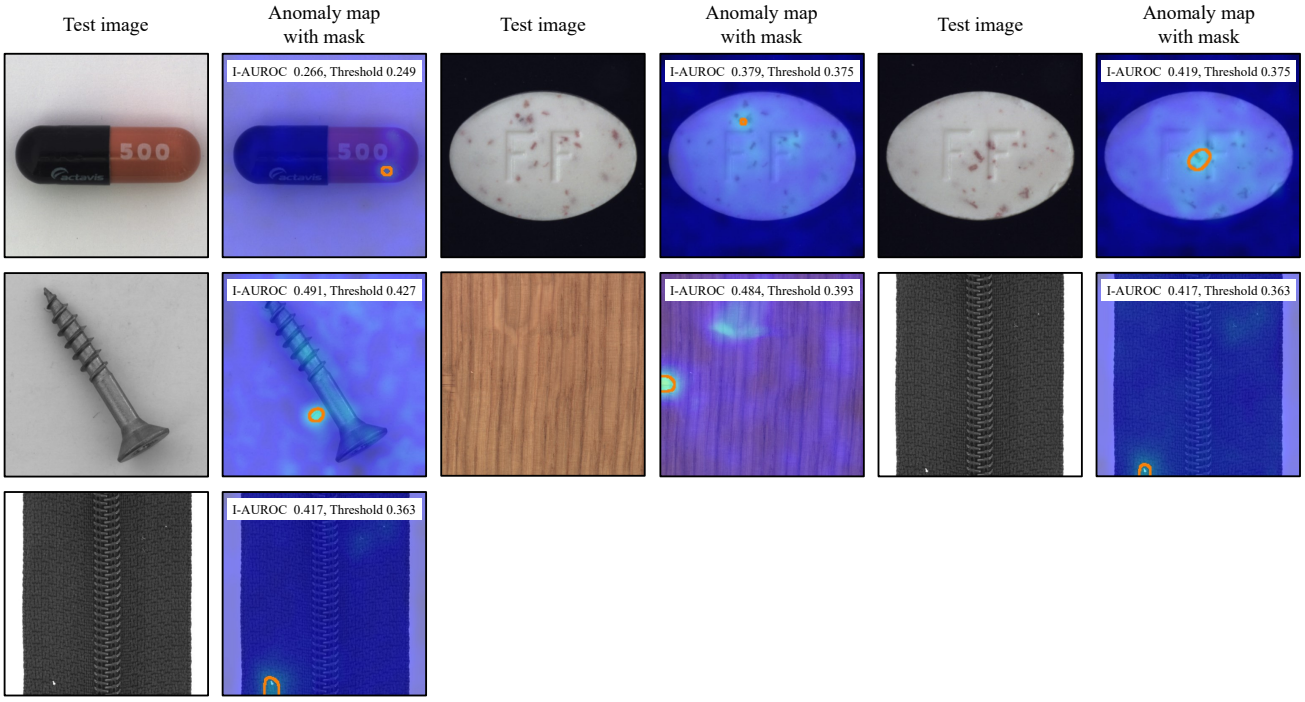

Figure S5: Visualization of all 7 false-positive classification cases on our proposed model (PNI). The contours overlaid on anomaly maps are from thresholds optimizing F1-scores of anomaly detection.

In Figure S5, we visualize false-positive images and the corresponding anomaly maps with masks, which are thresholded by the F1-optimal detection threshold. The main cause of false-positive errors is the variance of normal images. A stain in the capsule category in the first row of the first column, for example, is considered a normal pattern which is shown in the train dataset, but it is difficult to judge as normal since the stain pattern is various. In addition, the dust in the zipper category in the last row rarely appears in the train dataset, resulting in ambiguous labeling. To decrease false-positive errors, infrequent normal patterns, which are less likely to appear in the train dataset, should be trained with normal feature distribution, which leaves for further study.

In Figure S6, we visualize the false-negative images with the corresponding ground truth masks and the corresponding anomaly maps. Most false-negative errors are caused by detection failures of small anomaly patches. Since we extract local features from mid-level blocks of the pre-trained network, these small regions are concatenated with the neighborhood to generate local features, which could lead to insufficient weight to be judged the features as anomalies. In addition, small cracks in the pill category in the second row of Figure S6 are difficult to judge as abnormal since these patches are analogous to normal patches. To decrease false-negative errors, generating fine-grained local features for small patches are required. The boundary between normal and anomaly regions should be clearer through more advanced normal feature distribution.

Although those kinds of misclassified errors should be improved through further work, there are only 19 misclassified samples out of 1725 images and 7 categories are solved perfectly.

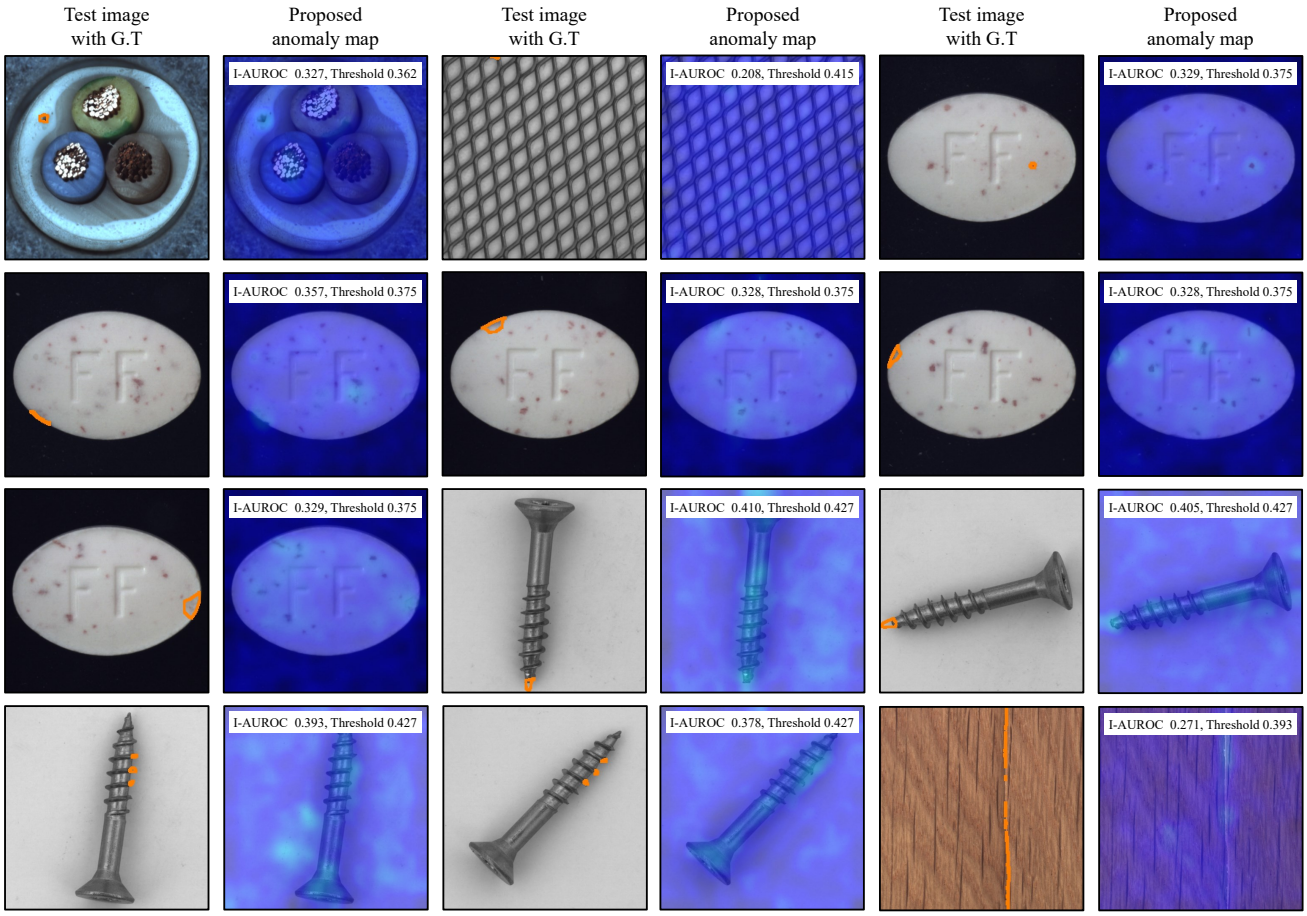

Figure S6: Visualization of all 12 false-negative classification cases on our proposed model (PNI). The contours overlaid on test images are the corresponding ground truth.

S5.2. Qualitative Comparison

To verify the effectiveness of our proposed model, we visualize some test images with the corresponding anomaly maps from both PatchCore [21] and our model in Figure S7. The proposed algorithm calculates anomaly maps closer to ground truth masks in various categories and cases.

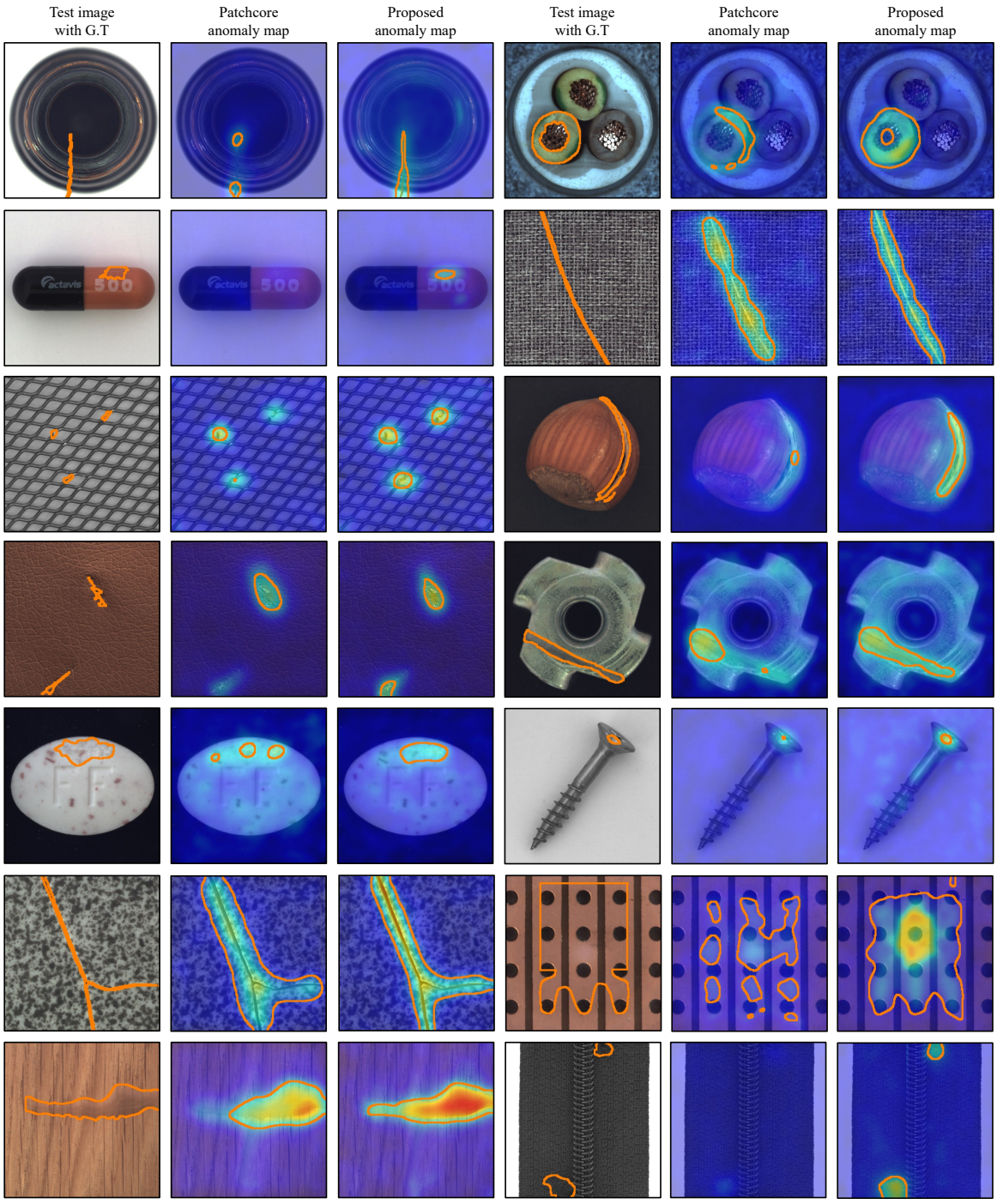

Figure S7: Comparison of the anomaly maps from our proposed model (PNI) and PatchCore on various classes of MVTec AD dataset. The contours overlaid on the anomaly maps are from thresholds optimizing F1-scores of anomaly detection.

A misplaced cable in the first row of the fourth column of Figure S7, for example, PatchCore cannot find appropriate anomalies since local features of misplaced cables are stored in the coreset of normal features. On the other hand, our proposed PNI evaluates the whole area of the misplaced cable as abnormal since the local features are incompatible with the corresponding position and neighborhood information. In addition, with the thread in the carpet image in the second row of the fourth column of Figure S7, our proposed PNI draws a more detailed and precise anomaly mask which is closer to ground truth, compared to PatchCore. This is because the PNI can refine anomaly maps with the trained refinement network to fit better with image patterns.

S5.3. Examples in BTAD

We visualize the test images with the ground truth masks and the corresponding anomaly maps with masks from BTAD [19] dataset in Figure S8, where all three categories are presented. The contours overlaid on the anomaly maps are from thresholds optimizing F1 scores of anomaly localization. We can find that the predicted masks generally follow the ground truth, which leads to the state-of-the-art anomaly localization performance, 97.8% P-AUROC.

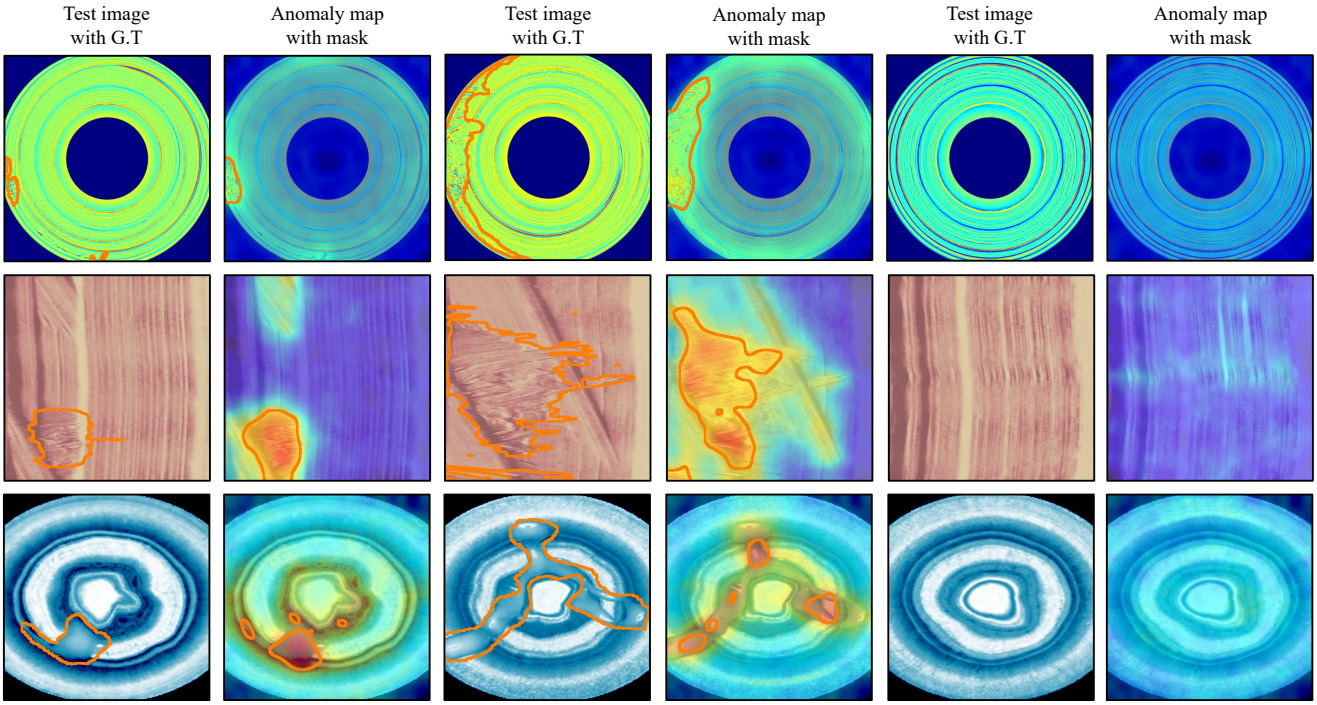

Figure S8: Examples of the test images (top), the anomaly maps (middle), and the predicted masks (bottom) on the BTAD. The ground truth anomaly masks are overlaid on test images, and the contours overlaid on the anomaly maps are from thresholds optimizing F1-scores of anomaly localization.

References

[S1] William Falcon et al. Pytorch lightning. *GitHub. Note: https://github. com/PyTorchLightning/pytorch-lightning*, 3(6), 2019.

[S2] Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with gpus. *IEEE Transactions on Big Data*, 7(3):535–547, 2019.

[S3] Philipp Liznerski, Lukas Ruff, Robert A Vandermeulen, Billy Joe Franks, Marius Kloft, and Klaus Robert Muller. Explainable deep one-class classification. 2021.

[S4] Ken Perlin. Making noise. *GDC Talk, 1999*, 1999.