TAB: Text-Align Anomaly Backbone Model for Industrial Inspection Tasks Supplementary Materials

1. Proposed method in details

1.1. Implementation Details

In terms of the implementation details for our anomaly backbone, all training images undergo resizing and center-cropping to dimensions of 256×256 and 224×224, respectively. We employ ResNet 18 as the primary backbone. During the training phase, we utilize an SGD optimizer with a learning rate of 1e-4. The training process spans 200 epochs with a batch size of 512, executed on an NVIDIA GeForce RTX 4090 with 24GB of memory.

For our CLIP model, we use the ViT-B/16 as the backbone, which is trained on the LAION-400M dataset [8]. Particularly, our text encoder adopts the transformer-based architecture present in the CLIP model, with specific parameters set at 512 for the dimensionality of the hidden state in the transformer, 8 heads, and 12 layers.

1.2. Algorithm

In this section, we elucidate our proposed Text-Align Anomaly Backbone (TAB) method, as referenced in the main paper and Algorithm 1. We ensure consistency in mathematical notations with those used in the main paper. The culmination of the pre-training flow is the weight of our convolutional neural network image encoder. We substitute the encoder weight of previous work with ours and faithfully re-implement prior work by adhering to their paper's training strategies and anomaly scoring functions, as delineated in the experiment section.

1.3. Similarity Matrix

In our proposed anomaly-text-aware pre-training strategy, we align visual image and text features, as depicted in Figure 4. The similarity matrix, represented in this figure, has rows corresponding to input batch anomaly images and columns corresponding to training class names, sentence groups, and averaged pooling text features. Higher and lower similarity scores are denoted by red and blue colors, respectively.

For instance, our objective is to enhance the similarity scores between the anomaly 'Bottle' image features and the 'Bottle' sentence features, while minimizing the similarity scores with other text feature groups. This is crucial because even without specific defect types for synthetic anomaly images, the image should exhibit higher similarity to corresponding abnormal text sentences containing the same class names. By considering both normal and abnormal similarity matrices during pre-training optimization, the model can effectively explore the diversity of normal and abnormal feature insights.

2. Prompt Samples

This section elucidates the association prompt phrases utilized in the Industrial Domain Prompt Association (IDPA), as detailed in Section 3.2 of the main paper. The two prompt states, namely 'prompt normal' and 'prompt abnormal', are delineated in Subsections 2.1 and 2.2, respectively. Subsection 2.3 describes the industrial domain prompt template. A prompt sentence is constructed by integrating a prompt template, a prompt state, and the name of the training class. The prompt template facilitates the replacement of the prompt state in the template, selecting from Subsection 2.1 for normal instances and Subsection 2.2 for abnormal instances. A few examples are provided below for further clarity:

- 'A photo of the damaged transistor.'
- 'A cropped photo of the transistor with damage.'

2.1. prompt normal

- 'good {class name}'
- 'unblemished {class name}'
- 'normal {class name}'
- 'perfect {class name}'
- 'good condition {class name}'
- '{class name} without flaw'
- '{class name} without defect'
- '{class name} without damage'

2.2. prompt abnormal

- 'not good {class name}'
- 'blemished {class name}'
- 'abnormal {class name}'
- 'imperfect {class name}'
- 'damaged {class name}'
- 'broken {class name}'
- '{class name} with flaw'
- '{class name} with defect'
- '{class name} with damage'
- '{class name} with distortion'
- '{class name} with stains'
- '{class name} with missing parts'
- '{class name} with broken parts'
- '{class name} with bumpy surfaces'
- '{class name} with dirty background'
- '{class name} with scratch'
- '{class name} with hole'

2.3. industrial domain prompt template

- 'a photo of a {prompt state}.'
- 'a photo of the {prompt state}.'
- 'an industrial photo of a {prompt state}.'
- 'an industrial photo of the {prompt state}.'
- 'an manufacturing photo of a {prompt state}.'
- 'an manufacturing photo of the {prompt state}.'
- 'a high reflection industrial photo of a {prompt state}.'
- 'a high reflection industrial photo of the {prompt state}.'
- 'an industrial photo of a rotated {prompt state}.'
- 'an industrial photo of the rotated {prompt state}.'
- 'a manufacturing cropped photo of a {prompt state}.'
- 'a manufacturing cropped photo of the {prompt state}.'
- 'a photo of the rotated {prompt state}.'
- 'a closed-up photo of the {prompt state}.'
- 'a blurry photo of the {prompt state}.'
- 'a photo of the small {prompt state}.'
- 'a photo of the big {prompt state}.'
- 'a low resolution photo of the {prompt state}.'
- 'a photo of one {prompt state}.'
- 'a photo of multiple {prompt state}.'

3. Additional qualitative results

3.1. Cross-dataset

In this study, we further demonstrate the generalization capabilities of our Anomaly Backbone weights, as shown in Figures 5 and 6. These weights are pre-trained on the MVTecAD[1] dataset and fine-tuned and tested on two other datasets, BTAD[7] and KSDD2[2]. Our proposed method, as depicted in Figure 5, effectively highlights anomalous regions with minimal false positives, particularly in classes 01 and 02 (rows 1 to 4). In Figure 6, our method significantly enhances RD4AD[4] by accurately localizing anomalous regions in industrial surface inspections, demonstrating the effectiveness and precision of our approach.

3.2. Defect classification

In the defect classification experiment, we utilize the MixedWM38 dataset (Figure 1). The dataset is split into a training set and a test set in an 8:2 ratio, and the backbone is trained in a supervised manner using the provided labels. We further illustrate the extracted test-set features for defect classification in Figure 2, demonstrating the capability of our weights and ImageNet to comprehend various defect types. For ease of visualization, we focus on eight basic defect types out of a total of 38.

As depicted in Figure 2, our initial model exhibits superior startup performance and understanding of varying wafer defects compared to ImageNet(initial), generating a distinct features cluster. After fine-tuning, Ours(Fine-tuned) provides a compact features cluster for each defect in the feature space, proving that our proposed weights offer superior insight and understanding than ImageNet in industrial defect classification scenarios.

3.3. Feature representation in latent space

In this section, we delve deeper into the influence of feature representation with varying backbone weights on the MVTecAD dataset, as depicted in Figure 7. These visualizations offer a more tangible understanding of the impact these factors have on the performance of our model, thereby providing valuable insights for future optimization and strategy development.

3.4. Industrial dataset

In this study, we introduce a meticulously curated manufacturing visual inspection dataset, referred to as the Industrial dataset. This dataset was constructed by selectively aggregating open-source datasets from global websites and official public conference workshops. The Industrial dataset encompasses 30 categories, including 15 object categories and 15 texture categories, amounting to a total of 17,393 images. Notably, the Industrial dataset exclusively comprises normal images. Representative image samples for each category are depicted in Figure 3.

4. Detailed quantitative results

In this section, we undertake a thorough analysis of the anomaly detection and localization results, with a specific focus on each category within the MVTecAD dataset. We draw insightful comparisons with established methodologies such as PaDiM[3] (ICPR'20), RD4AD[4] (CVPR'22), SimpleNet[6] (CVPR'23) and RegAD[5] (ECCV'22). The outcomes of this comparative study are systematically tabulated in Table 2 and 3, which concentrate on detection and localization metrics, respectively.

In addition, we broaden our analysis to encompass a detailed performance evaluation across the KSDD2[2] and BTAD[7] datasets. The specifics of this cross-dataset analysis are encapsulated in Table 4. This comprehensive examination not only deepens our understanding of the performance metrics across various categories but also illuminates potential areas of improvement for the evaluated methodologies.

The comparison of different augmentation techniques is mentioned in the main paper. We adopt the FID metric as the method to evaluate the quality of the generated image and compare it with the real defect images. As Table 1 indicates, NSA has the best-generated image quality with the lowest FID score.

	Mask	Perlin	CutPaste	NSA
FID	120.10	98.36	76.49	**64.00**

Table 1. Experiment of comparing the FID score of four synthetic methods.

References

[1] Paul Bergmann, Michael Fauser, David Sattlegger, and Carsten Steger. Mvtec ad–a comprehensive real-world dataset for unsupervised anomaly detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9592–9600, 2019. 2, 9, 10

[2] Jakob Božič, Domen Tabernik, and Danijel Skočaj. Mixed supervision for surface-defect detection: From weakly to fully supervised learning. *Computers in Industry*, 129:103459, 2021. 2, 3, 12

[3] Thomas Defard, Aleksandr Setkov, Angelique Loesch, and Romaric Audigier. Padim: a patch distribution modeling framework for anomaly detection and localization. In *International Conference on Pattern Recognition*, pages 475–489. Springer, 2021. 3, 9, 10

[4] Hanqiu Deng and Xingyu Li. Anomaly detection via reverse distillation from one-class embedding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9737–9746, 2022. 2, 3, 9, 10, 11, 12

[5] Chaoqin Huang, Haoyan Guan, Aofan Jiang, Ya Zhang, Michael Spratling, and Yan-Feng Wang. Registration based few-shot anomaly detection. In *European Conference on Computer Vision*, pages 303–319. Springer, 2022. 3, 9, 10

[6] Zhikang Liu, Yiming Zhou, Yuansheng Xu, and Zilei Wang. Simplenet: A simple network for image anomaly detection and localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20402–20411, 2023. 3, 9, 10

[7] Pankaj Mishra, Riccardo Verk, Daniele Fornasier, Claudio Piciarelli, and Gian Luca Foresti. Vt-adl: A vision transformer network for image anomaly detection and localization. In *2021 IEEE 30th International Symposium on Industrial Electronics (ISIE)*, pages 01–06. IEEE, 2021. 2, 3, 11

[8] Christoph Schuhmann, Richard Vencu, Romain Beaumont, Robert Kaczmarczyk, Clayton Mullis, Aarush Katta, Theo Coombes, Jenia Jitsev, and Aran Komatsuzaki. Laion-400m: Open dataset of clip-filtered 400 million image-text pairs. *arXiv preprint arXiv:2111.02114*, 2021. 1

Algorithm 1: Text-align Anomaly Backbone pre-training pseudo-code

```
# E_T:Text Encoder (Frozen)
# E_V:Visual Encoder (Trainable)
# labels:images class name list
# class_names:  training data class names
# X_n, X_a:normal, abnormal images
# T_n, T_a:normal, abnormal text sentences
# f_n^T, f_a^T:normal, abnormal text features
# f_n^V, f_a^V:normal, abnormal vision features
# L_CE:cross entropy loss
Input:  {X_n, class_names}
X_n = [X_1, X_2, ..., X_N]
class_names = [screw, cable, ...]
X_n, X_a = generate_synthetic(X_n)
T_n, T_a = generate_prompt(class_names)
f_n^T, f_a^T = E_T(T_n), E_T(T_a)
f_n^T, f_a^T = norm(f_n^T), norm(f_a^T)
for epoch in range(epochs):
   for (X_n, X_a labels) in data_loader:
      #
      f_n^V, f_a^V = E_V(X_n), E_V(X_a)
      # Normalize
      f_n^V, f_a^V = norm(f_n^V), norm(f_a^V)
      # Calculate Similarity Matrix, logits
      normal_logits = temperature * inner(f_n^V, f_n^T)
      abnormal_logits = temperature * inner(f_a^V, f_a^T)
      #
      loss_normal = L_CE(normal_logits, labels).mean()
      loss_abnormal = L_CE(abnormal_logits, labels).mean()
      #
      loss = (loss_normal + loss_abnormal)/2
      loss.backward()
   # end for
# end for
Output:  {E_V.weights}
```

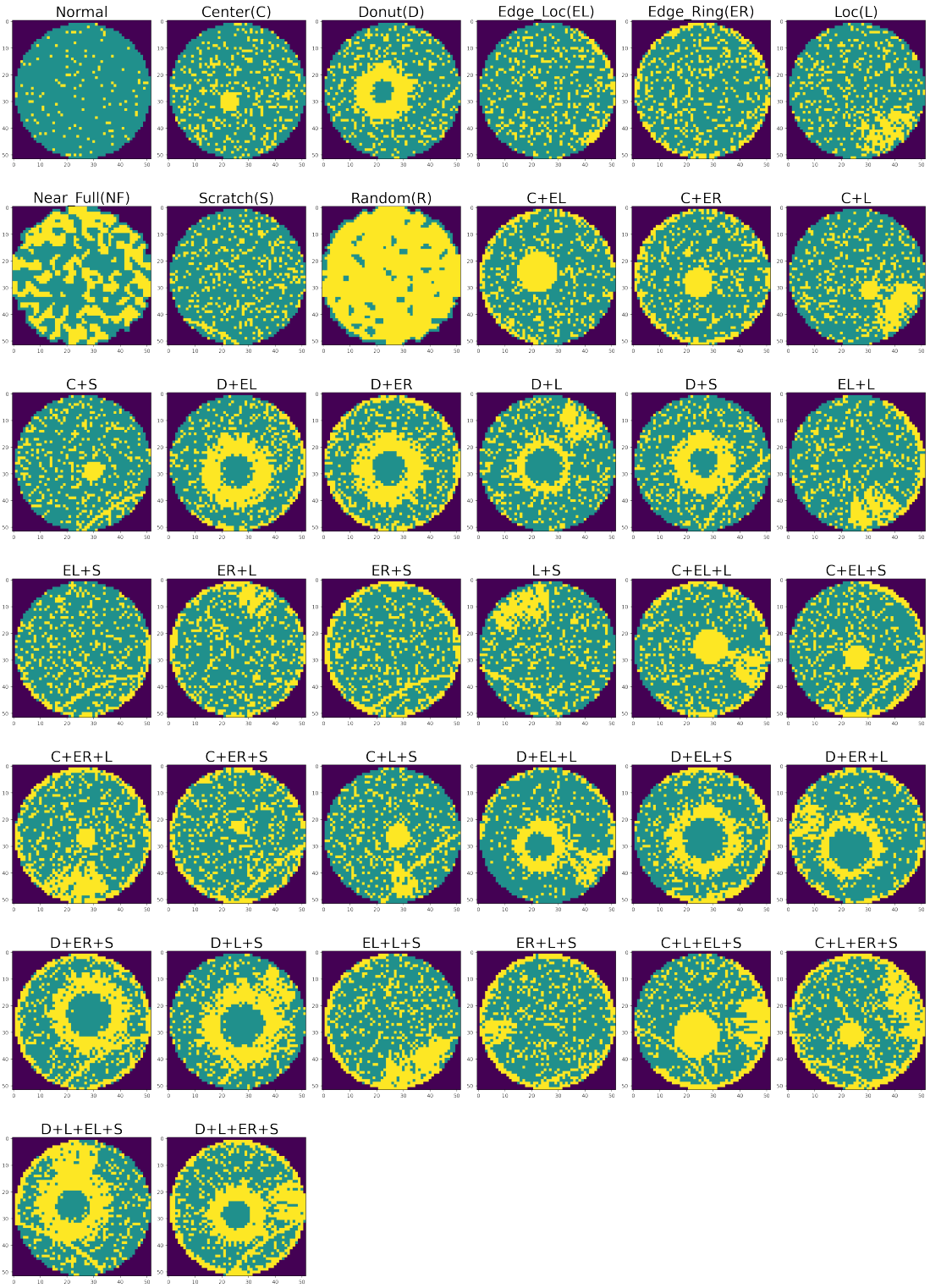

Figure 1. The samples of wafer MixedWM38 dataset. The MixedWM38 dataset is a mixed-type wafermap defect dataset formed by some real wafer patterns and synthetic samples. It contains 1 normal pattern, 8 basic single defect patterns, and 29 mixed defect patterns, a total of 38 defect patterns

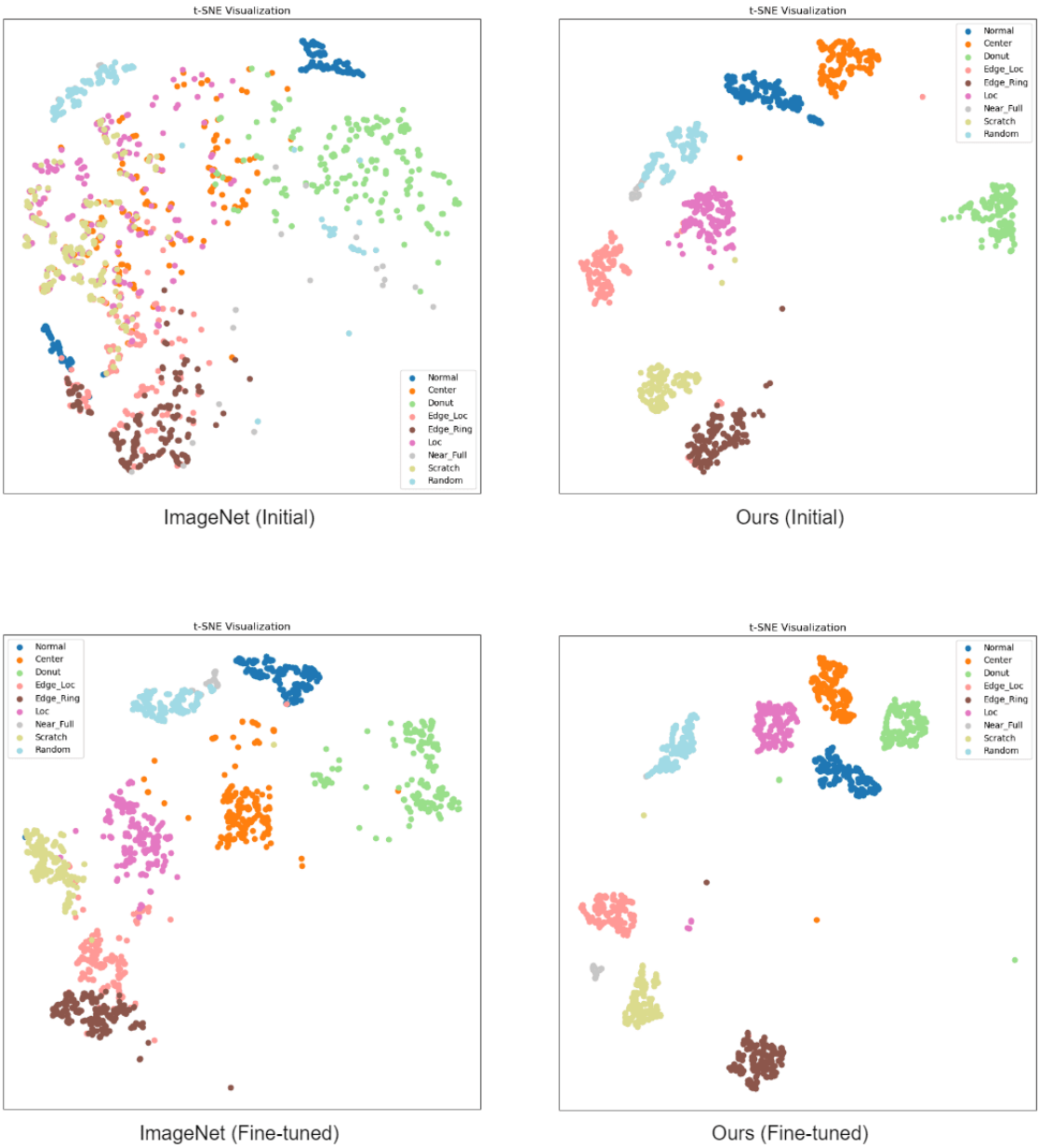

Figure 2. Visualization of features in the wafer MixedWM38 for defect classification.

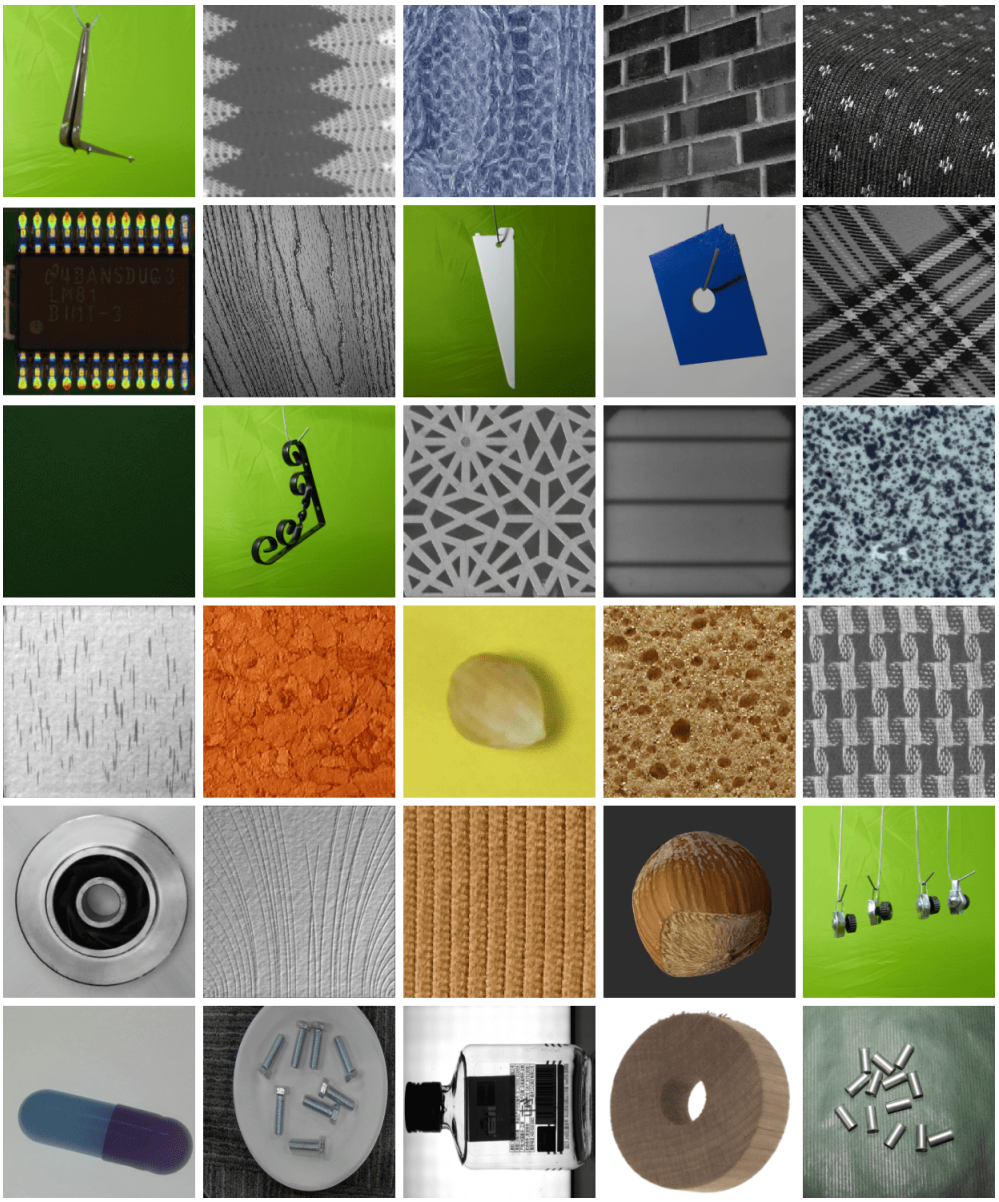

Figure 3. The samples of our well-organized Industrial dataset samples. It contains 30 categories, and only normal images are included.

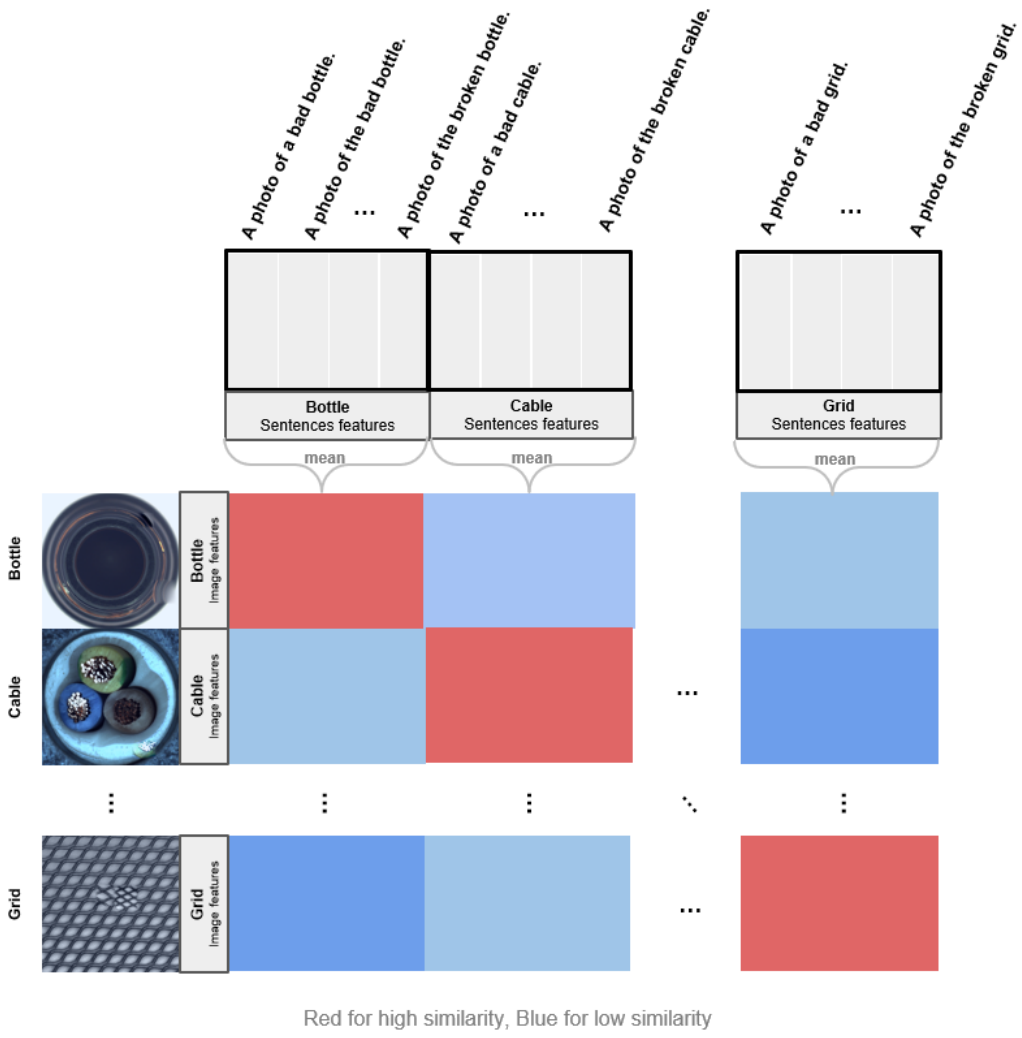

Figure 4. Similarity Matrix of images and text features are aligned in our Anomaly Backbone.

	PaDiM[3]		RD4AD[4]		SimpleNet[6]		RegAD(4shots)[5]		RegAD(8shots)[5]	
Pre-trained	ImgN.	TAB	ImgN.	TAB	ImgN.	TAB	ImgN.	TAB*	ImgN.	TAB*
carpet	98.40	**99.30**	**99.90**	99.85	**100.00**	97.57	97.90	**98.00**	**98.50**	98.20
grid	89.80	**93.19**	99.80	**100.00**	99.50	**99.77**	**91.20**	85.90	91.50	**91.70**
leather	98.90	**99.93**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**
tile	**95.90**	94.43	98.20	**99.37**	95.40	**96.93**	95.50	**98.70**	97.40	**98.80**
wood	99.00	**99.10**	99.30	**99.85**	**100.00**	99.93	98.60	**99.40**	**99.40**	**99.40**
Texture Avg.	96.40	**97.19**	99.44	**99.81**	**98.98**	98.84	**96.64**	96.40	97.36	**97.62**
bottle	**99.60**	99.23	**100.00**	**100.00**	**100.00**	**100.00**	99.40	**99.60**	**99.80**	**99.80**
cable	85.50	**88.23**	99.20	**99.52**	**99.00**	96.77	76.10	**76.70**	80.60	**83.30**
capsule	87.00	**87.80**	95.30	**95.50**	94.90	**95.77**	72.40	**75.20**	76.30	**78.00**
hazelnut	84.10	**85.17**	**99.90**	98.75	**99.29**	98.67	95.80	**97.40**	96.50	**97.70**
metal_nut	97.40	**98.83**	**100.00**	**100.00**	99.95	**100.00**	94.60	**97.60**	**98.30**	98.20
pill	86.90	**90.57**	97.00	**97.60**	93.90	**96.57**	80.80	**81.30**	80.60	**85.20**
screw	74.50	**77.37**	93.90	**95.01**	70.10	**90.57**	56.60	**61.80**	63.40	**70.30**
toothbrush	94.70	**96.83**	96.70	**98.25**	**100.00**	97.47	**90.90**	87.90	**98.50**	98.10
transistor	92.50	**93.80**	97.10	**97.20**	**97.70**	97.53	85.20	**87.50**	**93.40**	87.30
zipper	74.10	**85.83**	96.20	**96.31**	99.20	**99.63**	88.50	**91.00**	**94.00**	93.10
Object Avg.	87.63	**90.37**	97.53	**97.81**	95.40	**97.30**	84.03	**85.60**	88.14	**89.10**
Overall Avg.	90.55	**92.64**	98.17	**98.48**	96.60	**97.81**	88.23	**89.20**	91.21	**91.94**

Table 2. Comparison of TAB(Ours) and ImageNet pre-trained weights on SOTA methods for the image-level anomaly detection performance on MVTecAD[1] dataset. The abbreviation 'ImgN.' denotes 'ImageNet'. The results are reported in AUROC(%). Bold indicates the best performance. Please note that the asterisk (*) indicates using a pre-trained Industrial dataset to prevent overlap in the few-shot setting.

	PaDiM[3]		RD4AD[4]		SimpleNet[6]		RegAD(4shots)[5]		RegAD(8shots)[5]	
Pre-trained	ImgN.	TAB	ImgN.	TAB	ImgN.	TAB	ImgN.	TAB*	ImgN.	TAB*
carpet	**98.80**	98.63	**99.90**	99.48	86.80	**97.20**	**98.90**	98.70	**98.90**	98.70
grid	**93.60**	93.43	**99.20**	99.14	98.10	**98.40**	85.70	**87.20**	88.70	**89.40**
leather	**99.00**	98.97	99.20	**99.35**	98.56	**98.67**	99.10	**99.30**	98.90	**99.20**
tile	91.70	**93.20**	93.10	**96.30**	86.90	**91.77**	**94.90**	94.70	**95.20**	94.30
wood	**94.00**	93.83	**93.70**	93.40	89.90	**90.83**	**94.70**	94.60	94.60	**96.40**
Texture Avg.	95.42	**95.61**	97.02	**97.53**	92.05	**95.37**	94.66	**94.90**	95.26	**95.60**
bottle	98.10	**98.23**	98.50	**98.74**	97.02	**97.37**	98.40	**98.50**	97.50	**98.40**
cable	94.90	**95.60**	97.10	**98.42**	**93.90**	90.90	92.70	**93.50**	94.90	**96.10**
capsule	**98.20**	98.10	98.30	**99.15**	97.20	**98.37**	97.60	**98.00**	**98.20**	**98.20**
hazelnut	**97.90**	97.20	98.70	**99.60**	**95.21**	93.37	98.00	**98.10**	98.20	**98.70**
metal_nut	**96.70**	96.40	95.70	**97.83**	95.57	**96.40**	**97.80**	97.50	96.90	**98.50**
pill	94.60	**96.03**	97.30	**98.40**	97.20	**97.23**	**97.40**	96.70	97.80	**98.50**
screw	97.20	**97.27**	**99.20**	99.05	**97.30**	96.43	95.00	**95.50**	**97.10**	97.00
toothbrush	98.60	**98.73**	**99.00**	98.52	**98.10**	97.00	**98.50**	98.10	**98.70**	98.40
transistor	**96.80**	96.63	89.90	**96.01**	**90.20**	88.97	**93.80**	93.00	**96.80**	95.50
zipper	97.60	**98.07**	**98.90**	98.77	97.70	**98.77**	94.00	**98.40**	97.40	**98.20**
Object Avg.	97.06	**97.23**	97.26	**98.45**	**95.94**	95.48	96.32	**96.73**	97.38	**97.75**
Overall Avg.	96.51	**96.69**	97.18	**98.14**	94.64	**95.44**	95.77	**96.12**	96.67	**97.03**

Table 3. Comparison of TAB(Ours) and ImageNet pre-trained weights on SOTA methods for the pixel-level anomaly localization performance on MVTecAD[1] dataset. The abbreviation 'ImgN.' denotes 'ImageNet'. The results are reported in AUROC(%). Bold indicates the best performance. Please note that the asterisk (*) indicates using a pre-trained Industrial dataset to prevent overlap in the few-shot setting.

		Detection		Localization	
	Category	ImageNet	TAB	ImageNet	TAB
PaDiM (ICPR'20)	KSDD2	68.90	**70.30**	94.90	**95.50**
	01	**99.90**	99.10	**97.00**	95.60
	02	83.80	**84.70**	95.20	**95.30**
	03	99.30	**99.50**	**96.90**	95.60
	BTAD Avg.	94.33	**94.43**	**96.37**	95.50
SimpleNet (CVPR'23)	KSDD2	70.24	**70.47**	79.32	**88.04**
	01	96.99	**98.74**	**95.75**	94.66
	02	84.52	**85.08**	95.67	**96.44**
	03	98.51	**99.53**	93.08	**95.20**
	BTAD Avg.	93.34	**94.45**	94.83	**95.43**
RD4AD (CVPR'22)	KSDD2	92.50	**95.60**	97.30	**97.60**
	01	94.60	**95.57**	96.80	**96.85**
	02	77.40	**84.20**	96.80	**96.90**
	03	**96.60**	94.83	97.20	**97.45**
	BTAD Avg.	89.53	**91.53**	96.93	**97.07**

Table 4. Cross-dataset experiment was set up with a pre-training phase on all normal samples from the MVTecAD dataset, followed by testing on the KSDD2 and BTAD datasets. The BTAD Avg. refer to the mean of classes 01, 02, and 03.

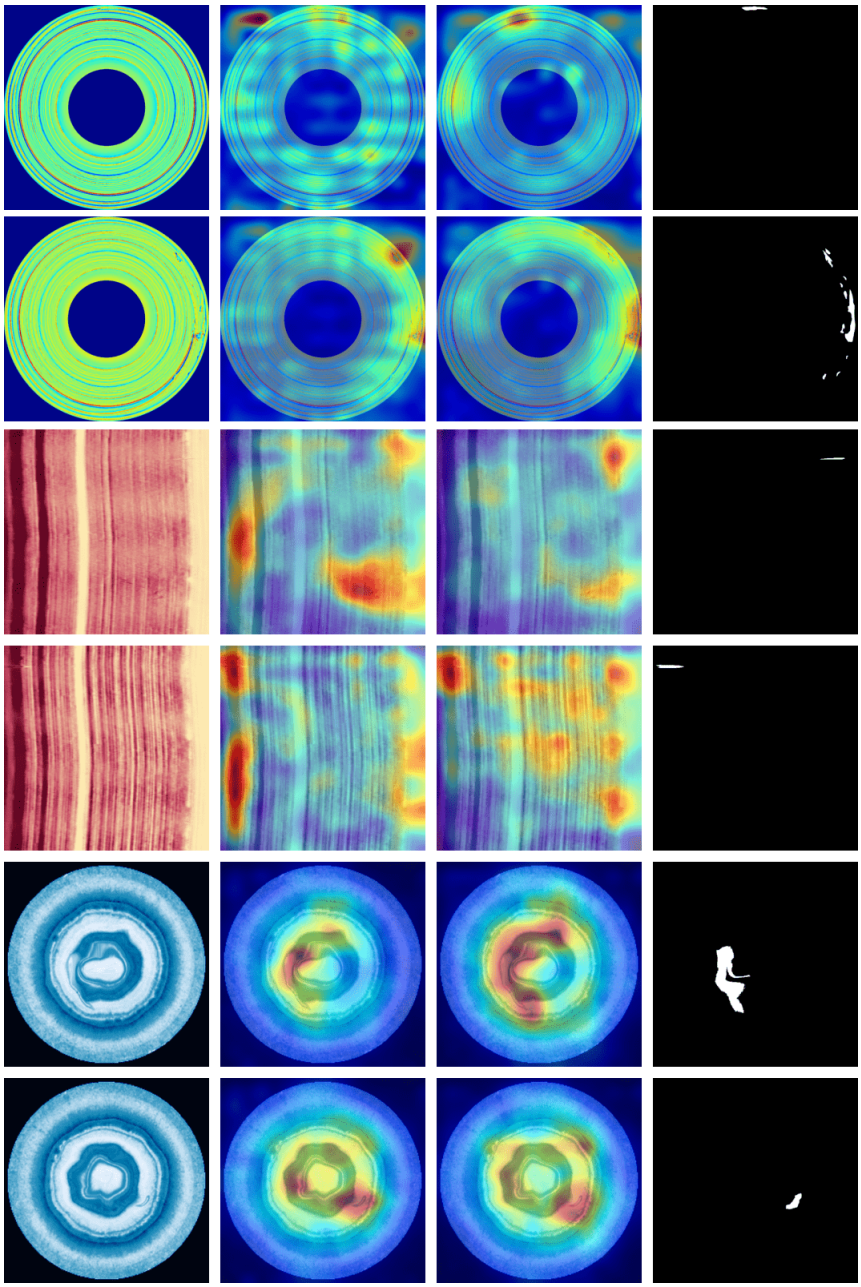

Figure 5. Visualization of RD4AD[4] localization results on BTAD dataset[7] with different backbone weights.From left to right column, Defect Image, ImageNet, Ours, Ground Truth respectively

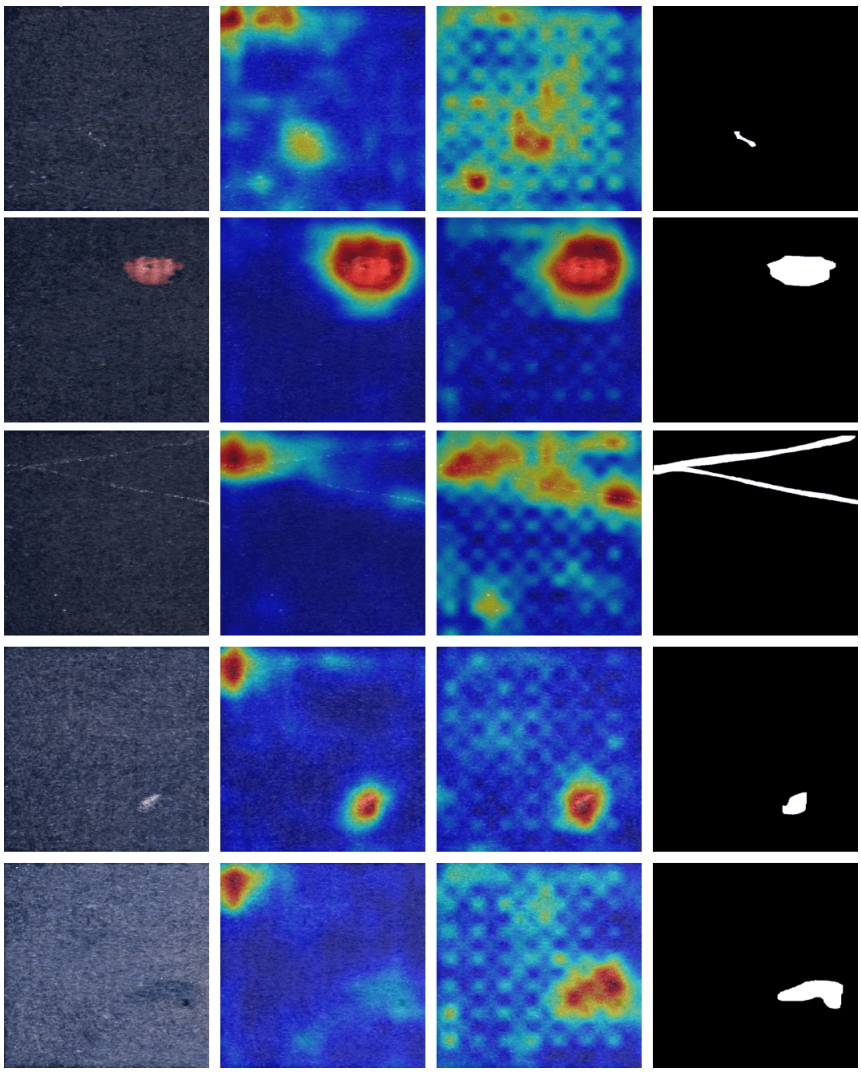

Figure 6. Visualization of RD4AD[4] localization results on KSDD2 dataset[2] with different backbone weights.From left to right column, Defect Image, ImageNet, Ours, Ground Truth respectively

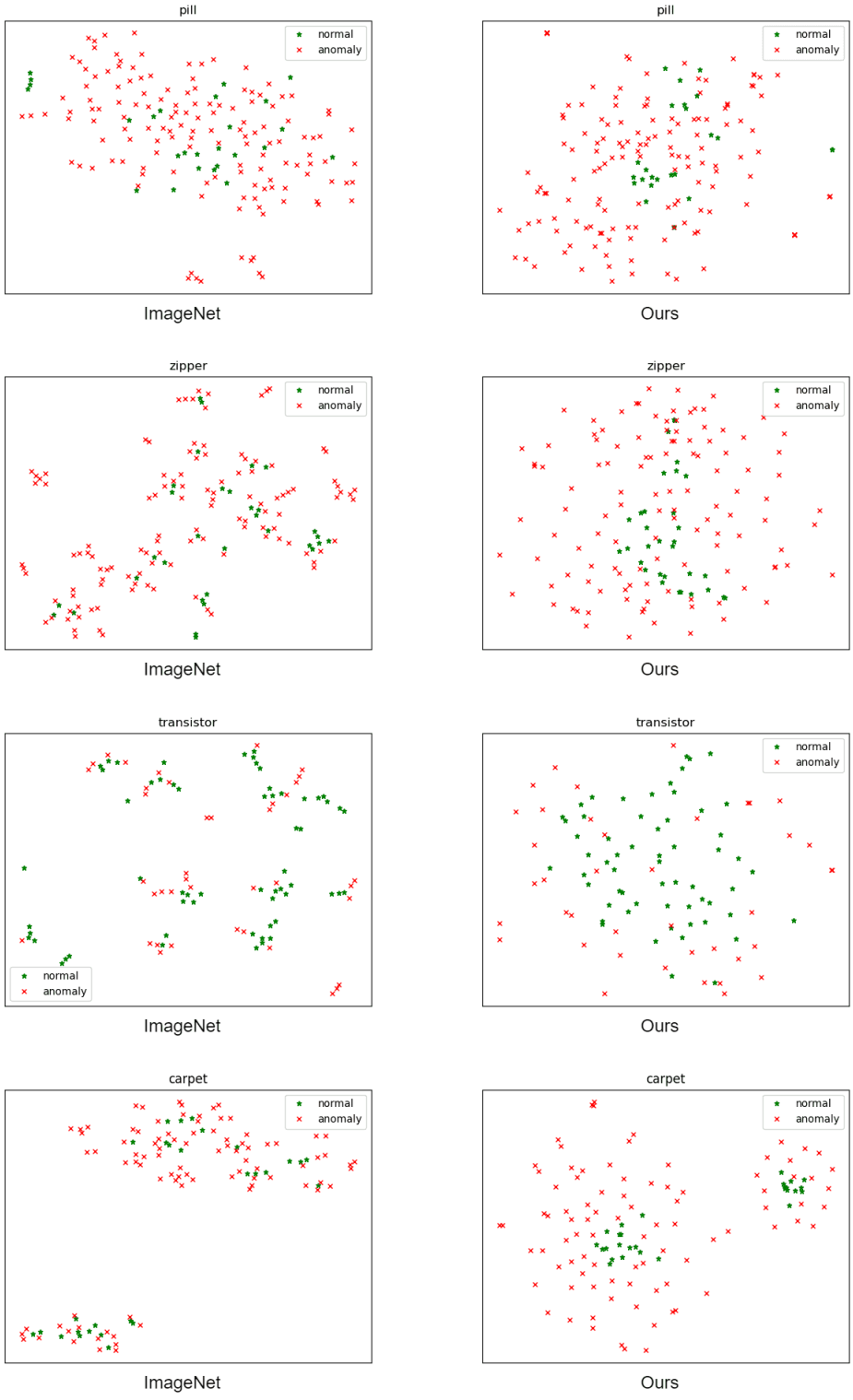

Figure 7. Visualization representation of ImageNet and TAB(Ours) on MVTecAD dataset. Ours can make feature representation discriminate between normal and anomaly features.