# Appendix for Bayesian Prompt Flow Learning for Zero-Shot Anomaly Detection

This appendix includes the following five parts: 1) Implementation details of our Bayes-PFL and the introduction of the state-of-the-art approaches in Section A; 2) Additional experimental results, including ablation studies, and further analysis in Section B; 3) Introduction to 15 industrial and medical datasets in Section C; 4) Presentation of more detailed quantitative and qualitative results in Section D; 5) Limitations of our method in Section E.

## A. Implementation Details and State-of-the-art Methods

### A.1. Implementation Details

**Details of the model architecture.** In Bayes-PFL, ViT-L-14-336 pretrained in CLIP [20] is adopted as the default backbone, with its image encoder consisting of 24 transformer layers. Following previous work [5, 6, 26], patch-level features at equal intervals from the image encoder (i.e., the 6th, 12th, 18th, and 24th layers) are extracted for fine-grained anomaly segmentation. Additionally, all input images are resized to a fixed resolution of $518 \times 518$ before being fed into the image encoder. Prior to obtaining the patch-level image features $\mathbf{F}^I$, a single linear layer without bias is first applied to map the raw patch features into the joint text-image space. In the prompt flow module, the flow length $K$ is set to 10 by default. The networks $f_\mu$ and $f_\Sigma$ used to generate the initial distribution's mean $\mu(\boldsymbol{\xi})$ and covariance $\Sigma(\boldsymbol{\xi})$, have the same MLP architecture, as illustrated by the prompt flow module in Figure 2 of the main text. Each network consists of three linear layers, with hidden layer dimensions equal to the joint embedding dimension $C$. SoftPlus activation function [22] is applied between every two consecutive layers. In the prompt banks, the number of learnable prompts in each bank is set to 3. The length of learnable context vectors $P$ and learnable state vectors $Q$ are both set to 5. During the auxiliary training and inference stages, the number of Monte Carlo sampling interations $R$ is set to 1 and 10, respectively.

**Details of sampling with prompt flow module.** The prompt flow module is used to approximate the Bayesian posterior distribution of the text prompt space using variational inference. As shown in Algorithm 1, it learns to map a simple initial distribution $q_0(\Phi_0)$ through a series of invertible linear transformations to a more complex distribution $q_K(\Phi_K)$ that closely approximates the true posterior. However, sampling from the distribution using Monte Carlo methods is a discrete process, which prevents the prompt flow module from optimizing its parameters through back-propagation. Therefore, reparameterization technique [15] is used, where two neural networks, $f_\mu$ and $f_\Sigma$, generate the mean $\mu(\boldsymbol{\xi})$ and covariance $\Sigma(\boldsymbol{\xi})$ of the initial normal distribution conditioned on $\boldsymbol{\xi}$, respectively. The symbol $\boldsymbol{\xi}$ represents either global image features or learnable free vectors. A single sample from the initial distribution is then acquired by computing: $p_r = \mu(\boldsymbol{\xi}) + \epsilon_r \odot \Sigma(\boldsymbol{\xi})$, where $\epsilon_r$ is obtained by randomly sampling from a standard normal distribution. The final sample $\boldsymbol{\varphi_r}$ can be obtained through $K$ invertible linear transformations as follows: $\boldsymbol{\varphi_r} = h_K(h_{K-1}(\cdots h_1(p_r))), r = 1, 2, \cdots, R$.

**Algorithm 1** Sampling with prompt flow module

**Input**: Condition input $\boldsymbol{\xi}$; Two neural networks with the same architecture $f_\mu$ and $f_\Sigma$; Invertible linear mapping $h_k, k = 1, 2, \cdots, K$; Number of Monte Carlo sampling interations $R$;

**Output**: Approximate posterior distribution of the prompt space $q_K(\Phi_K)$; The sampled prompts $\boldsymbol{\varphi_r}, r = 1, 2, \cdots, R$ from the distribution;

1: Generate the mean and covariance of the initial distribution $q_0(\Phi_0)$ conditioned on $\boldsymbol{\xi}$: $\mu(\boldsymbol{\xi}) = f_\mu(\boldsymbol{\xi})$, $\Sigma(\boldsymbol{\xi}) = f_\Sigma(\boldsymbol{\xi})$, and let $q_0(\Phi_0) = \mathcal{N}(\boldsymbol{\mu}(\boldsymbol{\xi}), \boldsymbol{\Sigma}(\boldsymbol{\xi}))$
2: Perform $K$ invertible linear mappings on the initial random vector $\Phi_0$ to obtain $\Phi_K = h_K(h_{K-1}(\cdots h_1(\Phi_0)))$. Then the final prompt distribution $q_K(\Phi_K)$ is acquired from Equation (6) of the main text.
3: **for** Interation $r = 1 : R$ **do**
4: Randomly sample $\epsilon_r$ from a standard normal distribution $\mathcal{N}(0, \mathbf{I})$;
5: Generate the initial sampled prompt using the reparameterization method: $p_r = \mu(\boldsymbol{\xi}) + \epsilon_r \odot \Sigma(\boldsymbol{\xi})$
6: Generate the final sampled prompts from $q_K(\Phi_K)$: $\boldsymbol{\varphi_r} = h_K(h_{K-1}(\cdots h_1(p_r)))$
7: **end for**
8: **return** $q_K(\Phi_K), \boldsymbol{\varphi_r}$

**Details of training and inference.** We conduct experi-

ments using seven datasets from the industrial domain and eight datasets from the medical domain. The industrial dataset VisA [27] is used as an auxiliary training set to fine-tune our Bayes-PFL. The resulting weights are directly applied to test on other industrial and medical datasets in a zero-shot manner. For VisA, the industrial dataset MVTec-AD [2] is adopted as the auxiliary training set. For the efficiency in auxiliary training phase, a single Monte Carlo sampling (i.e. $R = 1$) is used and the corresponding text embeddings are denoted as $\mathbf{Z}_{b,1}^t, b = 1, 2, \ldots, B$. At each training step, we randomly select one from the $B$ text embeddings to align with the image features. The final ZSAD result for each dataset is obtained by averaging the metrics of all categories contained within it. Note that during training, both the vanilla image and text encoders of CLIP are frozen, and only the parameters in newly added modules (e.g. the RCA module) are updated. This allows all of our experiments to be conducted on a single NVIDIA GeForce RTX 3090 with 24GB GPU memory. The optimizer used in the experiments is Adam [16], with an initial learning rate of 0.0001 and a batch size of 32, trained for 20 epochs. Five rounds of experiments are conducted using different random seeds, and the results are averaged to reduce experimental bias.

During the inference stage, $R = 10$ Monte Carlo samples are taken, and the corresponding text embeddings are denoted as $\mathbf{Z}_{b,r}^t, b = 1, 2, \cdots, B, r = 1, 2, \ldots, R$. These $B \times R$ text embeddings are each aligned with the image features, and the resulting anomaly maps and anomaly scores are simply integrated using averaging operation to obtain the final results. Note that when aligning with the patch-level features, we additionally employ the proposed RCA module to refine the text embeddings to $\mathbf{F}_{b,r}^t$

**Details of the loss function.** In Equation (14) of the main text, we state that the auxiliary training loss is derived as the sum of the orthogonal loss $\mathcal{L}_{ort}$ and the prompt flow loss $\mathcal{L}_p$. For $\mathcal{L}_{ort}$, it enforces the diversity of different prompts in the prompt banks through orthogonal constraints and can be easily computed from Equation (13) in the main text. Here, we provide a further explanation of $\mathcal{L}_p$ as introduced in Equation (7) of the main text. For convenience of explanation, we copy it from the main text below:

$$\mathcal{L}_p = E_{q_0(\Phi_0)}\left[\log q_0(\Phi_0) - \sum_{k=1}^{K} \log|1 + \mathbf{u}_k^T \phi(\Phi_k)|\right] - E_{q_0(\Phi_0)}[\log p(\Phi_K)] - E_{q_0(\Phi_0)}[\log p(D|\Phi_K)] \quad \text{(A.1)}$$

where $p = \mathcal{N}(0, \mathbf{I})$ and $q_0 = \mathcal{N}(\boldsymbol{\mu}(\boldsymbol{\xi}), \boldsymbol{\Sigma}(\boldsymbol{\xi}))$. The first two items can be easily obtained by calculating the log-likelihood for different sampled prompts under the assumption of a normal distribution. They ensure that the complex prompt distribution $q_K(\Phi_K)$ can be computed using a simple normal distribution $q_0(\Phi_0)$. For the third item, the goal is to maximize the log-likelihood of the auxiliary training data. This is approximated as the sum of the classification loss $\mathcal{L}_{cls}$ and the segmentation loss $\mathcal{L}_{seg}$:

$$\begin{aligned} -E_{q_0(\Phi_0)}[\log p(D|\Phi_K)] &= \mathcal{L}_{cls} + \mathcal{L}_{seg} \\ &= \mathcal{L}_{cls} + (\mathcal{L}_{focal} + \mathcal{L}_{dice}) \end{aligned} \quad \text{(A.2)}$$

where $\mathcal{L}_{cls}$ is a simple cross-entropy loss for classification. For $\mathcal{L}_{seg}$, the sum of the Focal Loss $\mathcal{L}_{focal}$ [21] and Dice Loss $\mathcal{L}_{dice}$ [17] for semantic segmentation is adopted. Specifically, the Focal Loss is to tackle the class imbalance between the background and anomalous regions, such as in the VisA [27] dataset, where the anomalous regions are significantly smaller than the background. It is computed as:

$$\mathcal{L}_{focal} = -\frac{1}{N}\sum_{i=1}^{N}(1 - p_i)^{\gamma}\log(p_i) \quad \text{(A.3)}$$

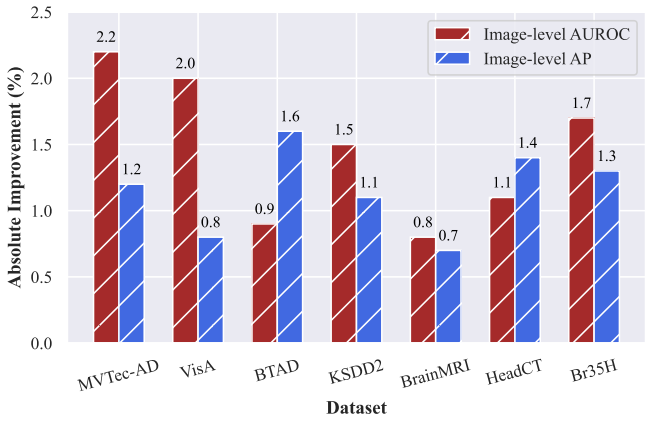
where $N$ is the number of pixels, $p_i$ is the probability predicted for the correct class. The symbol $\gamma$ is the focal factor, which adjusts the loss for easy and hard samples, and is set to 2 in this paper. Dice Loss quantifies the overlap between the predicted region and the ground truth, enabling the model to focus more on the anomalous areas. It is calculated as follows:

$$\mathcal{L}_{dice} = 1 - \frac{2\sum_i y_i \hat{y}_i}{\sum_i y_i + \sum_i \hat{y}_i} \quad \text{(A.4)}$$

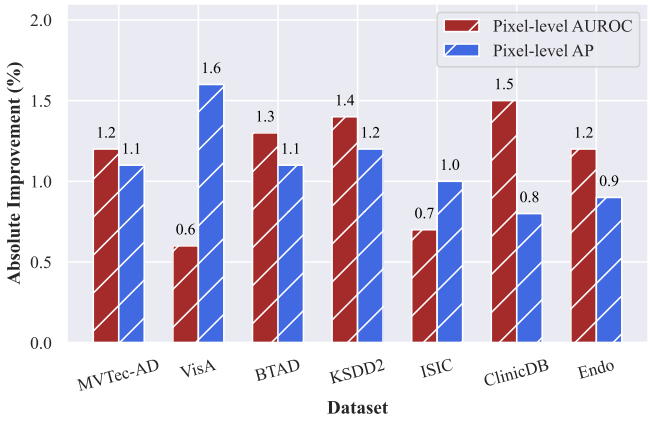
where $y_i$ and $\hat{y}_i$ represent the true label and predicted probability of the $i$-$th$ pixel in the image, respectively.

## A.2. State-of-the-art Methods

- **WinCLIP** [12] is one of the earliest works based on CLIP for the ZSAD task. Since the vanilla CLIP does not align text with fine-grained image features during pre-training, it addresses this limitation by dividing the input image into multiple sub-images using windows of varying scales. The final anomaly segmentation results are derived by harmoniously aggregating the classification outcomes of sub-images corresponding to the same spatial locations. In addition, a two-class text prompt design method, named Compositional Prompt Ensemble, is proposed and has been widely adopted in subsequent works.
- **APRIL-GAN** [6] adopts the handcrafted textual prompt design strategy from WinCLIP. However, for aligning textual and visual features, it introduces a linear adapter layer to project fine-grained patch features into a joint embedding space. After training on an auxiliary dataset, it can directly generalize to novel categories.
- **CLIP-AD** [7] builds upon the modality alignment design of APRIL-GAN, continuing to use a linear adapter to project patch-level image features. The key difference is that it incorporates a feature surgery strategy to further address the issues of opposite predictions and irrelevant highlights.




(a) Absolute improvement on the image-level metrics




(b) Absolute improvement on the pixel-level metrics

Figure A.1. Absolute performance improvement of our prompt-flow-based distribution modeling over Gaussian modeling in ZSAD. (a) Improvement of image-level metrics. (b) Improvement of pixel-level metrics. All values being positive indicates that our prompt-flow-based distribution outperforms the Gaussian distribution in the text prompt space.

Table A.1. The ZSAD performance when the single prompt (SP) and prompt ensemble (PE) are applied separately to the training and inference stages of the APRIL-GAN [6]. The best results are shown in **bold**.

| Training | | Inference | | Image-level | | Pixel-level | |
|---|---|---|---|---|---|---|---|
| SP | PE | SP | PE | AUROC | AP | AUROC | AP |
| ✔ | ✘ | ✔ | ✘ | 81.7 | 91.6 | 86.8 | 37.4 |
| ✔ | ✘ | ✘ | ✔ | 85.9 | 93.0 | 87.1 | 37.6 |
| ✘ | ✔ | ✔ | ✘ | 81.8 | 91.6 | 86.9 | 40.5 |
| ✘ | ✔ | ✘ | ✔ | **86.1** | **93.5** | **87.6** | **40.8** |

- **AnomalyCLIP** [26] employs a prompt-optimization-based text design strategy. By training on an auxiliary dataset, it learns object-agnostic text prompts that can be directly transferred to unseen object categories.
- **AdaCLIP** [5] introduces a hybrid prompt mechanism that integrates both dynamic and static prompts, embedding them into the text and image encoder layers. By incorporating visual prompts, the output text embeddings are able to dynamically adapt to the input image, thereby enhancing generalization performance.

Since the official code for WinCLIP has not been released, we use the reproduced code from [26]. For APRIL-GAN, CLIP-AD, AnomalyCLIP, and AdaCLIP, we retrained the models using the official code, maintaining the same backbone, input image resolution, and experimental settings (training on the VisA dataset and testing on other datasets) as those used in our Bayes-PFL. This ensures the fairness of the comparison between our Bayes-PFL and other state-of-the-art (SOTA) methods.

# B. Additional Results and Analysis

## B.1. Motivation

In Bayes-PFL, the text prompt space is modeled as a learnable distribution, and Monte Carlo sampling is used to sample from it to cover the prompt space. **A key question arises: what motivates this approach?** Our idea originally stemmed from the observation of using different types of manually designed prompts in the training and inference stages of APRIL-GAN [6]: single prompt (SP) or prompt ensemble (PE). For SP, the input text prompts are fixed, designed as *a photo of a perfect [class]* and *a photo of a damaged [class]* for normal and abnormal cases, respectively. For PE, the prompt templates and state words from the original paper are utilized to generate a large number of text prompts through combinations, as detailed in Section 2.2 of the main text. The experimental results, presented in Table A.1, reveal that using SP during both training and inference yields the worst ZSAD performance. Additionally, employing PE consistently outperforms the cases where it is not used, demonstrating its effectiveness.

**This inspires us that:** 1) using a single, fixed text prompt during training and inference hinders the model's generalization to novel categories; and 2) a richer and more diverse text prompt space can improve the model's ZSAD performance through the ensemble strategy. However, the number of manually designed text prompts is inherently limited, and their creation relies on expert knowledge, often requiring extensive trial and error. This observation motivates us to enhance the model's generalization performance on novel categories by learning a distribution over the prompt space, rather than relying solely on manually crafted prompts.

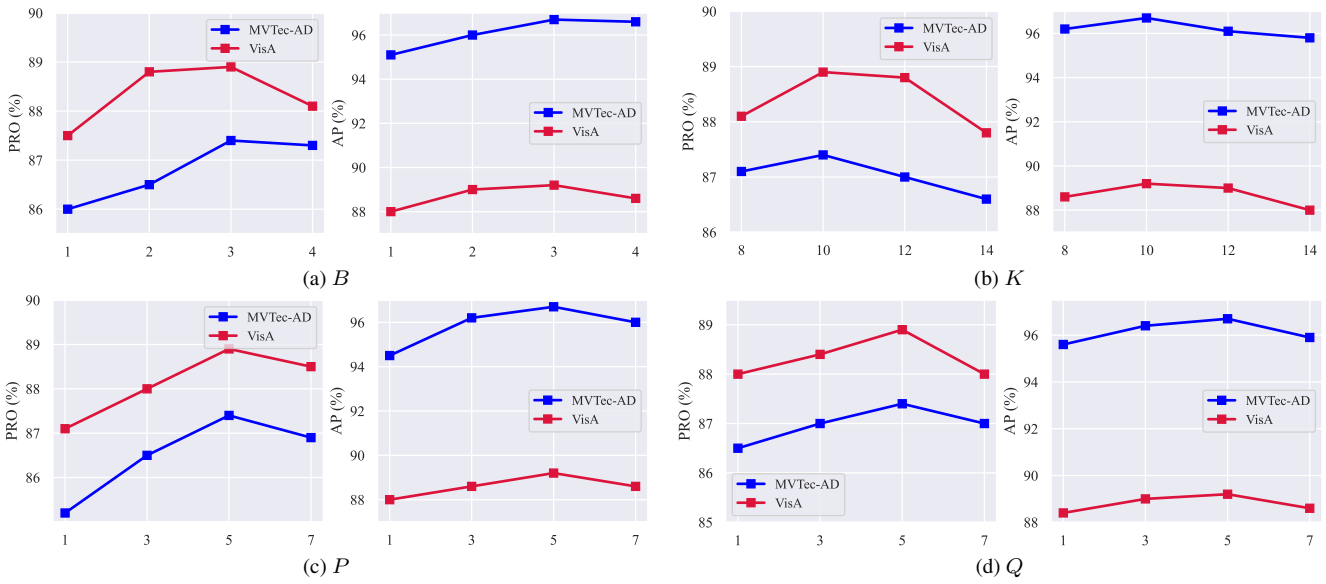



Figure A.2. (a) Ablation on the number of learnable prompts in each prompt bank $B$. (b) Ablation on the length of flow in the prompt flow module $K$. (c) Ablation on the length of learnable context vectors $P$. (d) Ablation on the length of learnable state vectors $Q$. Pixel/image-level (PRO, AP) performances are shown on the left and right sides of each subplot, respectively.

## B.2. Comparison of Different Distribution Types in the Text Prompt Sapce

In Bayes-PFL, the prompt flow module is designed to approximate the true posterior distribution of the prompt space. Specifically, Image-specific distribution (ISD) and Image-agnostic distribution (IAD) are used to model the context and state text prompt spaces, respectively. This design shifts the focus away from the specific form of the distribution, allowing us to directly learn how to transform a simple normal distribution into the target distribution in a data-driven manner using the auxiliary dataset.

**Is the distribution based on prompt flow better than other fixed-form distributions (e.g. Gaussian distributions)?** To investigate this, we model the text prompt space as a Gaussian distribution and compare it with our prompt-flow-based method. Specifically, the invertible linear transformation layer $h_k$ in the prompt flow module is removed, and the initial distribution $q_0 = \mathcal{N}(\boldsymbol{\mu}(\boldsymbol{\xi}), \boldsymbol{\Sigma}(\boldsymbol{\xi}))$ is directly used as the variational distribution to approximate the true posterior distribution of the prompt space. By substituting $q_0$ for $q_\gamma$ in Equation (5) of the main text, the new loss, which replaces $\mathcal{L}_p$, is computed as:

$$\mathcal{L}'_p = D_{KL}[q_0(\Phi)||p(\Phi)] - E_{q_0(\Phi)}[\log p(D|\Phi)] \quad \text{(A.5)}$$

where $D_{KL}$ denotes Kullback-Leibler divergence. The second term of Equation (A.5) is still approximated by the sum of classification and segmentation losses, which is similar to Bayes-PFL.

Figure A.1 illustrates the absolute performance improvement of our prompt-flow-based distribution modeling over Gaussian modeling in the text prompt space. Seven different datasets from industrial or medical domains are used to evaluate the two distinct distribution types. It can be observed that the image-level and pixel-level metrics across all datasets are positive, indicating that our method outperforms Gaussian modeling. This is attributed to our Bayes-PFL's ability to adaptively learn arbitrary complex distributions for the prompt space, rather than being constrained to a fixed Gaussian distribution.

## B.3. Additional Ablations

In this subsection, we perform additional ablation studies on hyperparameters as well as backbone and resolution settings.

**Ablation on hyperparameters.** As shown in Figure A.2, ablation experiments on hyperparameters are conducted on the MVTec-AD and VisA datasets, including (a) the number of learnable prompts in each prompt bank $B$, (b) the length of the flow in the prompt flow module $K$, (c) the length of learnable context vectors $P$, and (d) the length of learnable state vectors $Q$.

As observed in Figure A.2(a), the model achieves optimal ZSAD performance when the number of learnable prompts in the prompt bank is set to 3. This is because our auxiliary training data is relatively simple. Under the orthogonal loss constraint, three distinct learnable prompts, when fused with the distribution sampling results, are sufficient to capture anomalous semantics in the training set and generalize to novel categories. Figure A.2(b) demon-

Table A.2. Ablation on different bankbone and input image size.

| Backbone | Size | Image-level | | Pixel-level | |
|---|---|---|---|---|---|
| | | AUROC | AP | AUROC | AP |
| ViT-B-16-224 | $336^2$ | 87.0 | 93.6 | 90.7 | 39.9 |
| ViT-L-14-224 | $336^2$ | 92.1 | 96.0 | 91.6 | 44.1 |
| ViT-L-14-224 | $518^2$ | 89.7 | 95.3 | 91.5 | 44.6 |
| ViT-L-14-336 | $336^2$ | 91.7 | 96.2 | 90.9 | 44.1 |
| ViT-L-14-336 | $518^2$ | 92.3 | 96.7 | 91.8 | 48.3 |
| ViT-L-14-336 | $700^2$ | 90.1 | 94.3 | 90.2 | 46.4 |

Table A.3. Comparison of average inference time and maximum GPU cost on MVTec-AD. The best results are shown in **bold**.

| Method | Image-level | | Pixel-level | | GPU cost (GB) | Time (ms) |
|---|---|---|---|---|---|---|
| | AUROC | AP | AUROC | AP | | |
| WinCLIP | 91.8 | 95.1 | 85.1 | 18.0 | **2.0** | 840.0 |
| APRIL-GAN | 86.1 | 93.5 | 87.6 | 40.8 | 3.3 | **105.0** |
| CLIP-AD | 89.8 | 95.3 | 89.8 | 40.0 | 3.4 | 115.2 |
| AnomalyCLIP | 91.5 | 96.2 | 91.1 | 34.5 | 2.7 | 131.2 |
| AdaCLIP | 92.0 | 96.4 | 86.8 | 38.1 | 3.3 | 183.4 |
| Bayes-PFL | **92.3** | **96.7** | **91.8** | **48.3** | 3.3 | 388.5 |

strates that the model attains the highest pixel-level PRO and image-level AP when the length of the prompt flow, $K$, is set to 10 across both datasets. A flow that is too short fails to map the initial distribution to the approximate posterior of the prompt space. Conversely, when $K$ becomes too large, the generalization performance declines. This is due to the increased number of invertible linear transformations, leading to an overly complex learned prompt distribution that hinders transferability to new data domains. For the learnable context vector length $P$ in Figure A.2(c) and the state vector length $Q$ in Figure A.2(d), the model's performance first improves and then declines, with optimal performance observed when the length is set to 5. These learnable vectors function as biases when fused with the prompts sampled from the distribution, making the final generated prompts closer to the true word embedding space. A small vector length is insufficient to capture the semantics of context and state, while a large length introduces redundant information, which also hinders generalization.

**Influence of different backbones and resolutions.** In Table A.2, we analyze the effect of various pre-trained CLIP backbones and input image resolutions on the model's performance. The results indicate that larger backbones and appropriately higher input resolutions lead to more precise pixel-level segmentation. Notably, the ViT-L-14-336 backbone achieves the best ZSAD performance when the input image resolution is set to $518^2$. Consequently, we adopt this configuration as the default setting for our experiments. To ensure a fair comparison, we maintain consistent settings for other methods, including [5–7, 26].

## B.4. Additional Analysis

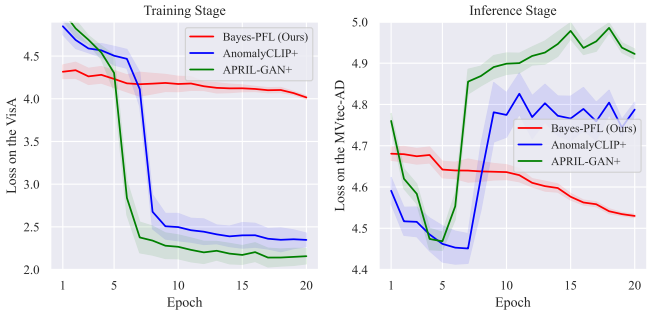
In this subsection, we evaluate the generalization performance and inference efficiency of various models. Furthermore, the text embedding is visualized to provide deeper insights into the proposed Bayes-PFL.

**Analysis of generalization performance.** In Section 4.2 of the main text, we compared the ZSAD performance of two improved methods, APRIL-GAN+ and AnomalyCLIP+, as well as our Bayes-PFL, across different training epochs. Experimental results show that the pixel-level AP of APRIL-GAN+ and AnomalyCLIP+ on the test set declines after reaching its peak. Here, we further investigate this phenomenon.

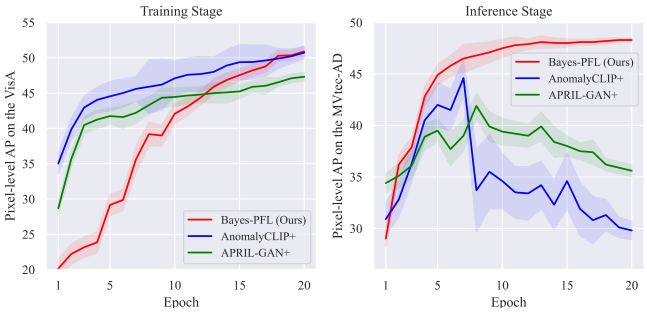
Figure A.3 shows the changes in both loss and pixel-level AP during the training and inference stages across different training epochs. APRIL-GAN+ and AnomalyCLIP+ demonstrate a consistent decrease in training loss, while the test loss decreases initially and then increases. In the training set, the AP steadily improves, whereas in the test set, the AP rises to a peak before exhibiting a downward trend. These trends suggest that both models gradually overfit the auxiliary training data, resulting in reduced generalization performance on unseen categories. In contrast, Bayes-PFL shows a gradually decreasing trend in test loss, with AP slowly increasing and eventually stabilizing during the inference stage. This indicates that our text prompt distributional strategy effectively mitigates overfitting to the training data, resulting in stronger zero-shot transfer capabilities.

**Analysis of inference efficiency.** Table A.3 compares the ZSAD performance of different methods, along with the maximum GPU consumption per image during inference and the average inference time per image on the MVTec-AD dataset. The proposed Bayes-PFL achieves faster inference speed than WinCLIP [12], but is slower than other methods. This is primarily due to the additional time required for sampling multiple text prompts from the distribution and performing inference. However, the ZSAD performance of Bayes-PFL significantly outperforms other methods, and this trade-off between inference speed and performance is deemed acceptable given its stronger generalization capability. Moreover, compared to APRIL-GAN and AdaCLIP, Bayes-PFL does not incur additional GPU memory consumption. This is because it avoids storing fixed text embeddings for all categories in memory, instead dynamically updating them based on the input image. The dynamic adjustment of text based on the input image enhances the model's anomaly detection capability for unseen categories to some extent.

**Analysis of the text embedding space.** In Bayes-PFL, we utilize the Image-Specific Distribution (ISD) and the Image-Agnostic Distribution (IAD) to model context words and state words in the text prompt space, respectively. Therefore, direct visualization of the prompt space is chal-




(a) **Left**: Loss on the VisA during the training stage. **Right**: Loss on the MVTec-AD during the inference stage




(b) **Left**: Pixel-level AP on the VisA during the training stage. **Right**: Pixel-level AP on the MVTec-AD during the inference stage

Figure A.3. (a) The variations in loss across different methods during training and inference stages. (b) The variations in pixel-level AP across different methods during training and inference stages.

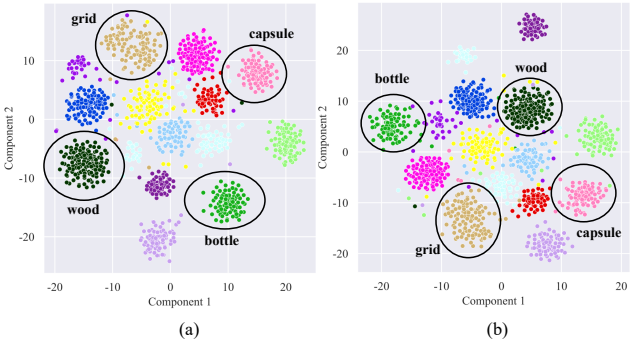



Figure A.4. The t-SNE visualization of text embeddings with two Monte Carlo samplings on the MVTec-AD dataset. (a) The first sampling. (b) The second sampling.

lenging. To address this, we employ t-SNE for dimensionality reduction to visualize the output of the text encoder, i.e., the text embeddings $\mathbf{Z}_{b,r}^{t}$. Figure A.4 presents the visualization results of text embeddings corresponding to abnormal prompts under two Monte Carlo sampling instances on the MVTec-AD dataset. Note that due to the introduction of ISD, each test image corresponds to a single data point in the picture. Two phenomena can be observed: 1) Text embeddings of the same category are clustered together, but differences still exist among individual image; 2) Different Monte Carlo sampling instances lead to corresponding changes in the text embeddings.

These observations align with our expectations. First, the introduction of visual features in ISD results in distinct prompt distributions for each test image, leading to differences in the text embeddings of individuals from the same category. However, due to the effective integration of visual semantics into the text prompts, text embeddings of the same category exhibit commonalities, causing them to cluster together. Second, the purpose of sampling from different distributions is to cover the prompt space as comprehensively as possible, thereby enhancing the generalization capability of the prompts for unseen categories. Consequently, different Monte Carlo sampling instances contribute to more diverse and enriched text embeddings, which improves the ZSAD performance.

## C. Datasets

In this section, we provide a brief introduction to the 15 datasets used in this work.

### C.1. Industrial Domain

- **MVTec-AD** [2] is specifically designed for industrial anomaly detection, consisting of 15 different categories (e.g., bottle, wood). In this work, we use only its labeled test set, which includes 467 normal images and 1,258 anomalous images. It also includes data from both texture and object types, making it more comprehensive, and is widely used in our ablation experiments.
- **VisA** [27] is a challenging industrial dataset that includes 12 categories (e.g., candle, capsules), all of which are object types. Its test set contains 962 normal images and 1,200 anomalous images, and is primarily used for auxiliary training in this study. Due to the small size of the anomalous regions relative to the background, it presents a significant challenge for both pixel-level classification and image-level segmentation.
- **BTAD** [18] includes three categories, all of which are object types, with resolutions ranging from 600 to 1600. The dataset contains 451 normal images and 290 anomalous images, used to evaluate the ZSAD performance.
- **KSDD2** [4] is a dataset designed for industrial defect detection. It includes 2,085 normal images and 246 abnormal images in the original training set, as well as 894 normal images and 110 abnormal images in the original test set. The image dimensions are similar, approximately 230 pixels in width and 630 pixels in height. In this work, we reconstruct the dataset for ZSAD. Specifically, all 356

abnormal images from the original training and test sets, along with an equal number of randomly selected normal images, are combined to form a new dataset. Note that this differs from the dataset processing approach in VCP-CLIP [19], where only all anomalous samples from the test set are used for zero-shot anomaly segmentation evaluation.

- **RSDD** [25] is a dataset for rail surface defect detection, consisting of two categories, both of which are object types. Its original dataset contains a total of 195 high-resolution abnormal images. To adapt it for the ZSAD task, the original images are cropped to generate 387 normal images and 387 abnormal images, each with a resolution of $512 \times 512$. Note that this differs from the dataset processing approach in VCP-CLIP [19], where only all anomalous samples from the test set are used for zero-shot anomaly segmentation evaluation.
- **DAGM** [9] is a texture dataset designed for weakly supervised anomaly detection, consisting of 10 categories. It contains 6,996 normal images and 1,054 abnormal images. Since the original pixel-level annotations are weak labels in the form of ellipses, we manually re-annotate the DAGM dataset for anomaly segmentation.
- **DTD-Synthetic** [1] is a synthetic dataset designed for texture anomaly detection, comprising 12 categories. It contains 357 normal images and 947 anomalous images.

### C.2. Medical Domain

- **HeadCT** [23] is a dataset for head CT scan analysis, containing images of brain scans with various conditions, such as hemorrhages and tumors, intended for anomaly detection. It includes a total of 100 normal images and 100 anomalous images, with image-level labels only. Therefore, it is used exclusively for the anomaly classification task. In this work, we directly adopt the dataset curated by AdaCLIP [5].
- **BrainMRI** [14] is a dataset for brain MRI analysis, comprising images of both healthy and abnormal brain scans, including conditions like tumors and lesions. It contains 98 anomalous images and 155 normal images, with only image-level labels. In this work, we directly adopt the dataset curated by AdaCLIP [5].
- **Br35H** [10] is a dataset for brain tumor detection in MRI images, containing 1500 normal images and 1500 anomalous images. Since it only provides image-level labels, it is used solely for anomaly classification task. In this work, we directly adopt the dataset curated by AdaCLIP [5].
- **ISIC** [8] is a dataset for skin lesion analysis in endoscopy images, containing a large collection of dermoscopic images, each labeled as either melanoma or non-melanoma lesions. It includes 379 anomalous images with pixel-level annotations, making it suitable only for anomaly segmentation task. In this work, we directly adopt the dataset curated by AdaCLIP [5].
- **CVC-ColonDB** [24] is a dataset for colorectal cancer detection in endoscopy images, containing colonoscopy images with labeled polyps. It includes a total of 380 anomalous images for colon polyp detection, with pixel-level annotations. Consequently, it is used for anomaly segmentation tasks in the medical domain in this work. In this work, we directly adopt the dataset curated by AdaCLIP [5].
- **CVC-ClinicDB** [3] is similar to CVC-ColonDB, containing 612 anomalous images with pixel-level annotations. Therefore, it is also used exclusively for anomaly segmentation tasks. In this work, we directly adopt the dataset curated by AdaCLIP [5].
- **Endo** [11] is another dataset similar to CVC-ColonDB, comprising 200 anomalous images with pixel-level annotations. While it is also used for colon polyp detection, differences in image acquisition devices and environments introduce a certain domain gap compared to other datasets. In this work, we directly adopt the dataset curated by AnomalyCLIP [26].
- **Kvasir** [13] is a larger medical dataset used for colon polyp detection in endoscopy images. It contains 1,000 anomalous images with pixel-level annotations and is used for anomaly segmentation task in the medical domain in this work. In this work, we directly adopt the dataset curated by AnomalyCLIP [26].

## D. Detailed ZSAD results

In Tables A.4 to A.9, we present the detailed quantitative results for each specific category of the MVTec-AD and VisA datasets. In Figures A.5 to A.34, we provide more extensive qualitative results for various categories across different industrial and medical datasets.

## E. Limitations

Our Bayes-PFL has already demonstrated the state-of-the-art ZSAD performance on 15 industrial and medical datasets. However, it still faces several limitations in practical applications: 1) During the inference stage, sampling from the learned prompt distribution incurs additional time overhead, hindering the model's ability to achieve optimal inference efficiency; 2) The vision-language model CLIP used in this study has limited capacity to understand textual information and extract fine-grained image features. In future work, we plan to investigate alternative vision-language models that may be better suited for fine-grained anomaly detection task.

Table A.4. Comparison of different categories in terms of pixel-level AUROC. The best results are shown in **bold**.

| Dataset | Category | WinCLIP | APRIL-GAN | CLIP-AD | AnomalyCLIP | AdaCLIP | Bayes-PFL |
|---|---|---|---|---|---|---|---|
| MVTec-AD | bottle | 89.5 | 83.5 | 91.2 | 90.4 | 83.8 | **94.0** |
| | cable | 77.0 | 72.2 | 76.2 | **78.9** | 85.6 | 78.4 |
| | capsule | 86.9 | 92.0 | 95.1 | 95.8 | 86.2 | **96.4** |
| | carpet | 95.4 | 98.4 | 99.1 | 98.8 | 94.8 | **99.6** |
| | grid | 82.2 | 95.8 | 96.3 | 97.3 | 90.6 | **98.2** |
| | hazelnut | 94.3 | 96.1 | 97.2 | 97.2 | 98.7 | **97.3** |
| | leather | 96.7 | 99.1 | 99.3 | 98.6 | 97.8 | **99.6** |
| | metal_nut | 61.0 | 65.5 | 58.9 | 74.6 | 55.4 | **77.5** |
| | pill | 80.0 | 76.2 | 83.7 | **91.8** | 77.5 | 89.1 |
| | screw | 89.6 | 97.8 | 98.7 | 97.5 | **99.2** | 98.4 |
| | tile | 77.6 | 92.7 | 94.5 | **94.7** | 83.9 | 92.8 |
| | toothbrush | 86.9 | 95.8 | 92.7 | 91.9 | 93.4 | **93.7** |
| | transistor | 74.7 | 62.4 | **75.5** | 70.8 | 71.4 | 67.5 |
| | wood | 93.4 | 95.8 | 96.9 | 96.4 | 91.2 | **97.1** |
| | zipper | 91.6 | 91.1 | 92.8 | 91.2 | 91.8 | **97.4** |
| | mean | 85.1 | 87.6 | 89.8 | 91.1 | 86.8 | **91.8** |
| VisA | candle | 88.9 | 97.8 | 98.7 | 98.8 | 98.6 | **99.2** |
| | capsules | 81.6 | 97.5 | 97.4 | 94.9 | 96.1 | **98.7** |
| | cashew | 84.7 | 86.0 | 91.4 | 93.7 | 97.2 | **91.6** |
| | chewinggum | 93.3 | 99.5 | 99.2 | 99.2 | 99.2 | **99.6** |
| | fryum | 88.5 | 92.0 | 93.0 | 94.6 | 93.6 | **94.7** |
| | macaroni1 | 70.9 | 98.8 | 98.7 | 98.3 | 98.8 | **99.3** |
| | macaroni2 | 59.3 | 97.8 | 97.6 | 97.6 | 98.2 | **98.3** |
| | pcb1 | 61.2 | 92.7 | 92.6 | 94.0 | 90.7 | **93.8** |
| | pcb2 | 71.6 | 89.8 | 91.0 | 92.4 | 91.3 | **92.4** |
| | pcb3 | **85.3** | 88.4 | 87.5 | 88.3 | 87.7 | 88.4 |
| | pcb4 | 94.4 | 94.6 | 95.9 | **95.7** | 94.6 | 95.3 |
| | pipe_fryum | 75.4 | 96.0 | 96.9 | **98.2** | 95.7 | 95.4 |
| | mean | 79.6 | 94.2 | 95.0 | 95.5 | 95.1 | **95.6** |

Table A.5. Comparison of different categories in terms of image-level AUROC. The best results are shown in **bold**.

| Dataset | Category | WinCLIP | APRIL-GAN | CLIP-AD | AnomalyCLIP | AdaCLIP | Bayes-PFL |
|---|---|---|---|---|---|---|---|
| MVTec-AD | bottle | **99.2** | 92.0 | 96.4 | 88.7 | 95.6 | 95.6 |
| | cable | **86.5** | 88.2 | 80.4 | 70.3 | 79.0 | 81.5 |
| | capsule | 72.9 | 79.8 | 82.8 | 89.5 | 89.3 | **92.4** |
| | carpet | 100.0 | 99.4 | 99.5 | 99.9 | 100.0 | **100** |
| | grid | 98.8 | 86.2 | 94.1 | 97.8 | 99.2 | **99.7** |
| | hazelnut | 93.9 | 89.4 | 98.0 | 97.2 | 95.5 | **95.9** |
| | leather | 100.0 | 99.7 | 100.0 | 99.8 | 100.0 | **100** |
| | metal_nut | **97.1** | 68.2 | 75.1 | 92.4 | 79.9 | 75.9 |
| | pill | 79.1 | 80.8 | 87.7 | 81.1 | **92.6** | 82.2 |
| | screw | 83.3 | 85.1 | 89.1 | 82.1 | 83.9 | **89.4** |
| | tile | **100.0** | 99.8 | 99.6 | 100 | 99.7 | 99.3 |
| | toothbrush | 87.5 | 53.2 | 76.1 | 85.3 | **95.2** | 90.5 |
| | transistor | 88.0 | 80.9 | 79.3 | **93.9** | 82 | 84.5 |
| | wood | **99.4** | 98.9 | 98.9 | 96.9 | 98.5 | 98.1 |
| | zipper | 91.5 | 89.4 | 88.6 | 98.4 | 89.4 | **99.7** |
| | mean | 91.8 | 86.1 | 89.8 | 91.5 | 92.0 | **92.3** |
| VisA | candle | 95.4 | 82.5 | 89.4 | 80.9 | **95.9** | 92.3 |
| | capsules | 85.0 | 62.3 | 75.2 | 82.7 | 81.1 | **92.1** |
| | cashew | **92.1** | 86.7 | 83.7 | 76.0 | 89.6 | 91.3 |
| | chewinggum | 96.5 | 96.5 | 95.6 | 97.2 | **98.5** | 97.5 |
| | fryum | 80.3 | 93.8 | 78.7 | 92.7 | 89.5 | **95.8** |
| | macaroni1 | 76.2 | 69.5 | 80.0 | 86.7 | 86.3 | **90.8** |
| | macaroni2 | 63.7 | 65.7 | 67.0 | 72.2 | 56.7 | **67.6** |
| | pcb1 | 73.6 | 50.6 | 68.6 | 85.2 | **74.0** | 66.1 |
| | pcb2 | 51.2 | 71.6 | 69.7 | 62.0 | 71.1 | **76.5** |
| | pcb3 | 73.4 | 66.9 | 67.3 | 61.7 | 75.2 | **79.0** |
| | pcb4 | 79.6 | 94.6 | 96.2 | 93.9 | 89.6 | **96.3** |
| | pipe_fryum | 69.7 | 89.4 | 86.5 | 92.3 | 88.8 | **98.7** |
| | mean | 78.1 | 78.0 | 79.8 | 82.1 | 83.0 | **87.0** |

Table A.6. Comparison of different categories in terms of pixel-level AP. The best results are shown in **bold**.

| Dataset | Category | WinCLIP | APRIL-GAN | CLIP-AD | AnomalyCLIP | AdaCLIP | Bayes-PFL |
|---|---|---|---|---|---|---|---|
| MVTec-AD | bottle | 49.8 | 53.0 | 56.8 | 55.3 | 49.8 | **67.4** |
| | cable | 6.2 | 18.2 | 17.3 | 12.3 | 16.5 | **20.7** |
| | capsule | 8.6 | 29.6 | 27.2 | 27.7 | 24.8 | **32.6** |
| | carpet | 25.9 | 67.5 | 65.4 | 56.6 | 63.5 | **84.5** |
| | grid | 5.7 | 36.5 | 30.7 | 24.1 | 27.8 | **41.3** |
| | hazelnut | 33.3 | 49.7 | 59.2 | 43.4 | **69.5** | 56.9 |
| | leather | 20.4 | 52.3 | 50.5 | 22.7 | 53.6 | **63.7** |
| | metal_nut | 10.8 | 25.9 | 21.2 | 26.4 | 19.9 | **27.7** |
| | pill | 7.0 | 23.6 | 26.1 | **34.1** | 25.8 | 30.5 |
| | screw | 5.4 | 33.7 | 39.1 | 27.5 | **41.6** | 40.0 |
| | tile | 21.2 | 66.3 | 65.2 | 61.7 | 48.8 | **76.6** |
| | toothbrush | 5.5 | **43.2** | 29.9 | 19.3 | 24.7 | 29.4 |
| | transistor | **20.2** | 11.7 | 14.2 | 15.6 | 11.9 | 13.3 |
| | wood | 32.9 | 61.8 | 59.4 | 52.6 | 56.6 | **73.2** |
| | zipper | 19.4 | 38.7 | 38.5 | 38.7 | 36.0 | **66.5** |
| | mean | 18.0 | 40.8 | 40.0 | 34.5 | 38.1 | **48.3** |
| VisA | candle | 2.4 | 29.9 | 36.6 | 25.6 | 45.3 | **38.5** |
| | capsules | 1.4 | 40.0 | 38.5 | 29.3 | 18.2 | **47.2** |
| | cashew | 4.8 | 15.1 | 24.1 | 19.6 | **44.8** | 25 |
| | chewinggum | 24.0 | 83.6 | 83.4 | 56.3 | **87.6** | 83.8 |
| | fryum | 11.1 | 22.1 | 22.4 | 22.6 | 24.0 | **28.5** |
| | macaroni1 | 0.03 | 24.8 | 23.2 | 14.9 | **27.1** | 23.9 |
| | macaroni2 | 0.02 | **6.8** | 2.3 | 1.5 | 3.0 | 4.0 |
| | pcb1 | 0.4 | 8.4 | 7.2 | **8.6** | 7.8 | 7.6 |
| | pcb2 | 0.4 | 15.4 | 8.2 | 9.1 | 17.5 | **17.9** |
| | pcb3 | 0.7 | 14.1 | 11.7 | 4.3 | 16.1 | **19** |
| | pcb4 | 15.5 | 24.9 | 31.2 | 30.6 | 34.2 | **31.9** |
| | pipe_fryum | 4.4 | 23.6 | 27.2 | **33.2** | 24.4 | 31.2 |
| | mean | 5.0 | 25.7 | 26.3 | 21.3 | 29.2 | **29.8** |

Table A.7. Comparison of different categories in terms of image-level AP. The best results are shown in **bold**.

| Dataset | Category | WinCLIP | APRIL-GAN | CLIP-AD | AnomalyCLIP | AdaCLIP | Bayes-PFL |
|---|---|---|---|---|---|---|---|
| MVTec-AD | bottle | 98.3 | 97.7 | 98.8 | 96.8 | 98.6 | **98.7** |
| | cable | 86.2 | **92.9** | 88.9 | 81.7 | 87.3 | 90.1 |
| | capsule | 93.4 | 95.4 | 96.4 | 97.8 | 97.8 | **98.4** |
| | carpet | 99.9 | 99.8 | 99.8 | 99.9 | 100.0 | **100** |
| | grid | 99.8 | 94.9 | 97.9 | 99.3 | 99.7 | **99.9** |
| | hazelnut | 96.3 | 94.6 | **99.0** | 98.5 | 97.5 | 98.0 |
| | leather | 100.0 | 99.9 | 100.0 | 99.9 | 100.0 | **100** |
| | metal_nut | 97.9 | 91.8 | 94.4 | **98.1** | 95.6 | 94.6 |
| | pill | **96.5** | 96.1 | 97.6 | 95.3 | 98.6 | 95.9 |
| | screw | 88.4 | 93.6 | 96.2 | 92.9 | 93.0 | **96.0** |
| | tile | 99.9 | 99.9 | 99.8 | **100** | 99.9 | 99.7 |
| | toothbrush | 96.7 | 71.9 | 90.2 | 93.9 | 97.9 | **96.8** |
| | transistor | 74.9 | 77.6 | 73.7 | **92.1** | 83.8 | 82.3 |
| | wood | 98.8 | 99.6 | 99.6 | 99.2 | 99.5 | **99.7** |
| | zipper | 98.9 | 97.1 | 96.9 | 99.5 | 97.1 | **99.9** |
| | mean | 95.1 | 93.5 | 95.3 | 96.2 | 96.4 | **96.7** |
| VisA | candle | 95.6 | 85.9 | 91.6 | 82.6 | **96.4** | 93.8 |
| | capsules | 80.9 | 74.6 | 86.6 | 89.4 | 86.7 | **95.9** |
| | cashew | 95.2 | 93.9 | 92.4 | 89.3 | 95.4 | **96.3** |
| | chewinggum | 98.8 | 98.4 | 98.1 | 98.8 | 99.4 | **99.5** |
| | fryum | 92.5 | 97.0 | 90.4 | 96.6 | 95.1 | **98.4** |
| | macaroni1 | 64.5 | 67.5 | 81.1 | 85.5 | 85.0 | **92.2** |
| | macaroni2 | 65.2 | 64.9 | 65.3 | **70.8** | 54.3 | 70.4 |
| | pcb1 | 74.6 | 54.6 | 72.5 | **86.7** | 73.5 | 70.0 |
| | pcb2 | 44.2 | 73.8 | 71.4 | 64.4 | 71.6 | **77.3** |
| | pcb3 | 66.2 | 70.5 | 71.9 | 69.4 | 77.9 | **81.1** |
| | pcb4 | 70.1 | 94.8 | 96.0 | 94.3 | 89.8 | **96.1** |
| | pipe_fryum | 82.1 | 94.6 | 93.7 | 96.3 | 93.9 | **99.4** |
| | mean | 77.5 | 81.4 | 84.3 | 85.4 | 84.9 | **89.2** |

Table A.8. Comparison of different categories in terms of pixel-level PRO. The best results are shown in **bold**.

| Dataset | Category | WinCLIP | APRIL-GAN | CLIP-AD | AnomalyCLIP | AdaCLIP | Bayes-PFL |
|---|---|---|---|---|---|---|---|
| MVTec-AD | bottle | 76.4 | 45.6 | 71.7 | 80.8 | 26.9 | **87.6** |
| | cable | 42.9 | 25.7 | 51.4 | 64.0 | 15.2 | **69.1** |
| | capsule | 62.1 | 51.3 | 62.6 | 87.6 | 65.7 | **93.4** |
| | carpet | 84.1 | 48.5 | 83.0 | 90.0 | 19.6 | **98.6** |
| | grid | 57.0 | 31.6 | 67.8 | 75.4 | 46.2 | **93.5** |
| | hazelnut | 81.6 | 70.3 | 83.5 | **92.5** | 42.3 | 87.1 |
| | leather | 91.1 | 72.4 | 95.5 | 92.2 | 55.9 | **99.1** |
| | metal_nut | 31.8 | 38.4 | 72.7 | 71.1 | 20.7 | **74.9** |
| | pill | 65.0 | 65.4 | 87.5 | 88.1 | 37.0 | **93.8** |
| | screw | 68.5 | 67.1 | 88.5 | 88.0 | 75.3 | **93.0** |
| | tile | 51.2 | 26.7 | 61.1 | 87.4 | 7.7 | **90.3** |
| | toothbrush | 67.7 | 54.5 | 83.5 | **88.5** | 25.6 | 87.6 |
| | transistor | 43.4 | 21.3 | 34.9 | **58.2** | 6.7 | 55.5 |
| | wood | 74.1 | 31.1 | 85.6 | 91.5 | 58.3 | **95.1** |
| | zipper | 71.7 | 10.7 | 30.2 | 65.4 | 3.4 | **90.1** |
| | mean | 64.6 | 44.0 | 70.6 | 81.4 | 33.8 | **87.4** |
| VisA | candle | 83.5 | 92.5 | 94.4 | **96.5** | 71.6 | 95.6 |
| | capsules | 35.3 | 86.7 | 87.2 | 78.9 | 80.3 | **87.4** |
| | cashew | 76.4 | 91.7 | 90.6 | **91.9** | 45.6 | 90.5 |
| | chewinggum | 70.4 | 87.3 | 82.7 | **90.9** | 53.9 | 87.4 |
| | fryum | 77.4 | 89.7 | 87.5 | 86.8 | 55.6 | **90.4** |
| | macaroni1 | 34.3 | 93.2 | 93.8 | 89.7 | 86.6 | **96.5** |
| | macaroni2 | 21.4 | 82.3 | 83.9 | 83.9 | 84.8 | **89.3** |
| | pcb1 | 26.3 | 87.5 | 84.1 | 80.7 | 52.3 | **89.3** |
| | pcb2 | 37.2 | 75.6 | 77.7 | 78.2 | 77.5 | **78.3** |
| | pcb3 | 56.1 | 77.8 | 78.8 | 76.8 | 76.2 | **79.3** |
| | pcb4 | 80.4 | 86.8 | 88.9 | **89.4** | 84.3 | 88.6 |
| | pipe_fryum | 82.3 | 90.9 | 93.2 | **96.1** | 86.3 | 95.3 |
| | mean | 56.8 | 86.8 | 86.9 | 87.0 | 71.3 | **88.9** |

Table A.9. Comparison of different categories in terms of image-level F1-max. The best results are shown in **bold**.

| Dataset | Category | WinCLIP | APRIL-GAN | CLIP-AD | AnomalyCLIP | AdaCLIP | Bayes-PFL |
|---|---|---|---|---|---|---|---|
| MVTec-AD | bottle | **97.6** | 92.8 | 94.6 | 90.9 | 93.7 | 94.6 |
| | cable | **84.5** | 84.5 | 79.3 | 77.4 | 79.2 | 79.8 |
| | capsule | 91.4 | 92.0 | 91.1 | 91.7 | 91.8 | **95.2** |
| | carpet | 99.4 | 98.3 | 99.4 | 99.4 | 100 | **100** |
| | grid | 98.2 | 89.1 | 92.3 | 97.3 | 98.2 | **99.1** |
| | hazelnut | 89.7 | 87.0 | **95.6** | 92.7 | 93.7 | 92.3 |
| | leather | 100.0 | 98.9 | 100.0 | 99.5 | 100 | **100** |
| | metal_nut | **96.3** | 89.4 | 89.4 | 93.6 | 89.4 | 89.4 |
| | pill | 91.6 | 91.6 | 92.1 | 92.1 | 93.7 | **93.9** |
| | screw | 87.4 | 88.9 | 90.3 | 88.3 | 89.2 | **90.8** |
| | tile | **99.4** | 98.8 | 98.8 | 100.0 | 98.8 | 98.3 |
| | toothbrush | 87.9 | 83.3 | 84.8 | **90.0** | 96.7 | 88.9 |
| | transistor | 79.5 | 76.1 | 69.6 | **83.7** | 77.1 | 78.4 |
| | wood | **98.3** | 96.8 | 96.7 | 96.6 | 96.7 | 96.8 |
| | zipper | 92.9 | 90.8 | 90.4 | 97.9 | 91.1 | **99.2** |
| | mean | 92.9 | 90.4 | 91.1 | 92.8 | 92.7 | **93.1** |
| VisA | candle | 89.4 | 76.9 | 82.2 | 75.6 | **90.2** | 85.4 |
| | capsules | 83.9 | 78.1 | 77.9 | 82.2 | 82.8 | **89.1** |
| | cashew | 88.4 | 85.7 | 83.7 | 80.3 | 87.5 | **89.4** |
| | chewinggum | 94.8 | 93.2 | 92.8 | 94.8 | 96.0 | **96.4** |
| | fryum | 82.7 | 91.8 | 81.4 | 90.1 | 87.0 | **93.9** |
| | macaroni1 | 74.2 | 70.8 | 77.4 | 80.4 | 80.0 | **82.1** |
| | macaroni2 | 69.8 | 69.3 | 69.7 | 71.2 | 67.8 | **69.9** |
| | pcb1 | 71.0 | 66.9 | 68.5 | **78.8** | 72.1 | 66.9 |
| | pcb2 | 67.1 | 69.7 | 70.9 | 67.8 | 72.5 | **73.3** |
| | pcb3 | 71.0 | 66.7 | 68.1 | 66.4 | 71.9 | **75.7** |
| | pcb4 | 74.9 | 87.3 | 91.3 | 87.8 | 82.1 | **91.4** |
| | pipe_fryum | 80.7 | 88.1 | 86.6 | 89.8 | 88.5 | **95.4** |
| | mean | 79.0 | 78.7 | 79.2 | 80.4 | 81.6 | **84.1** |

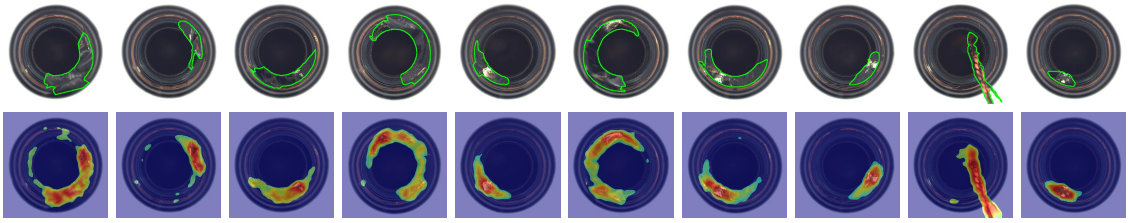


Figure A.5. Visualization of segmentation results for the bottle category on MVTec-AD. The first row represents the input images, with green outlines indicating the ground truth regions. The second row shows the results of anomaly segmentation.

Figure A.6. Visualization of segmentation results for the capsule category in MVTec-AD. The first row represents the input images, with green outlines indicating the ground truth regions. The second row shows the results of anomaly segmentation.

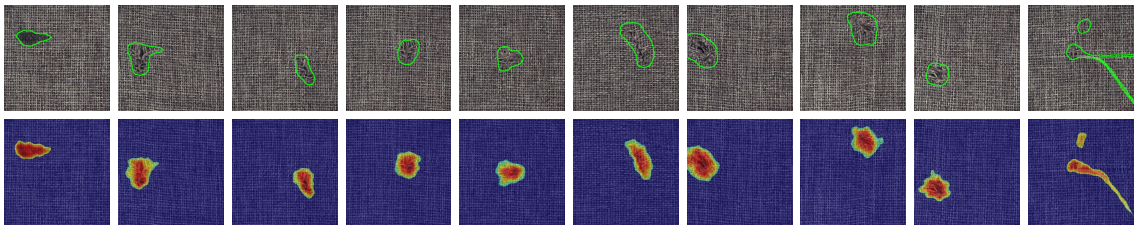


Figure A.7. Visualization of segmentation results for the carpet category in MVTec-AD. The first row represents the input images, with green outlines indicating the ground truth regions. The second row shows the results of anomaly segmentation.

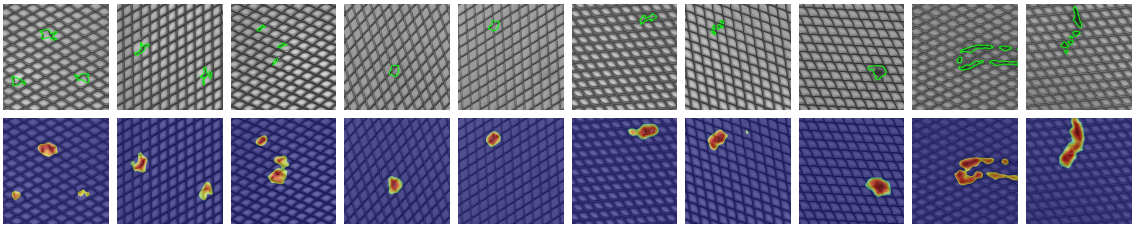


Figure A.8. Visualization of segmentation results for the grid category in MVTec-AD. The first row represents the input images, with green outlines indicating the ground truth regions. The second row shows the results of anomaly segmentation.

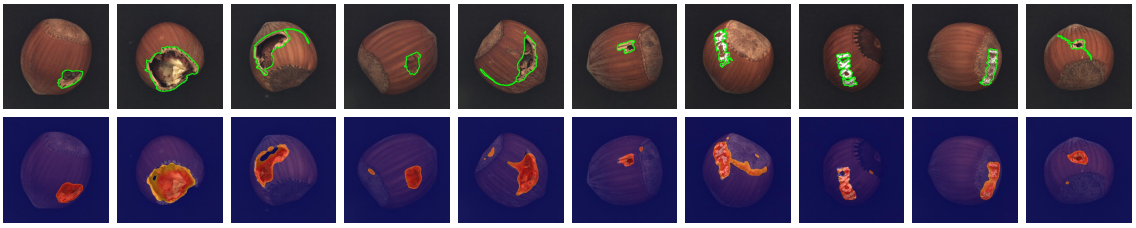


Figure A.9. Visualization of segmentation results for the hazelnut category in MVTec-AD. The first row represents the input images, with green outlines indicating the ground truth regions. The second row shows the results of anomaly segmentation.

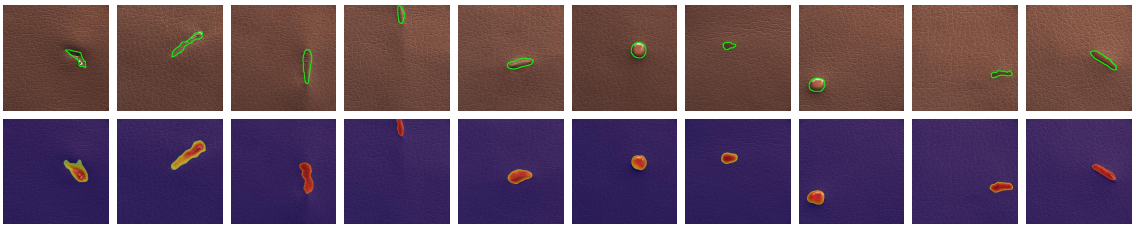


Figure A.10. Visualization of segmentation results for the leather category in MVTec-AD. The first row represents the input images, with green outlines indicating the ground truth regions. The second row shows the results of anomaly segmentation.

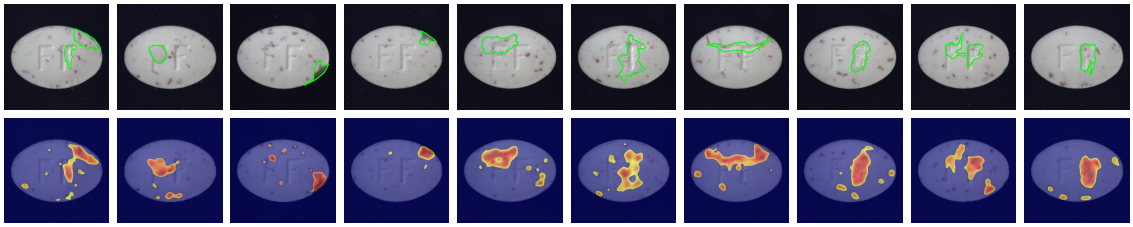


Figure A.11. Visualization of segmentation results for the pill category on MVTec AD. The first row represents the input images, with green outlines indicating the ground truth regions. The second row shows the results of anomaly segmentation.

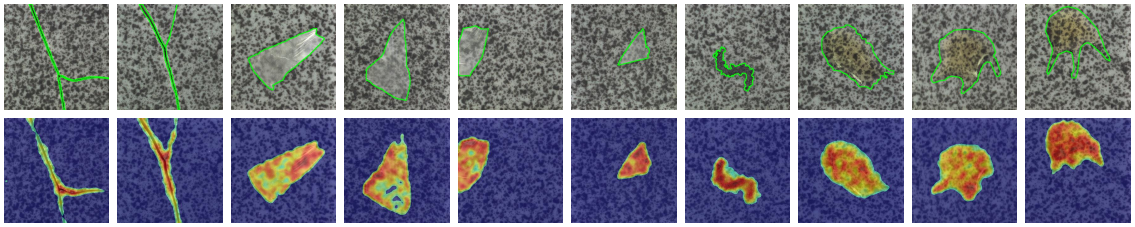


Figure A.12. Visualization of segmentation results for the tile category on MVTec-AD. The first row represents the input images, with green outlines indicating the ground truth regions. The second row shows the results of anomaly segmentation.

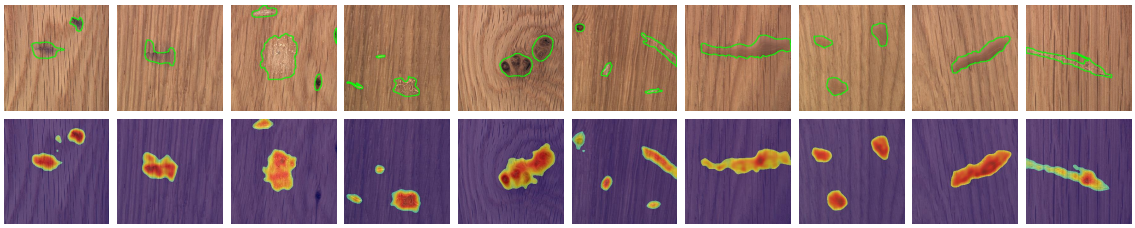


Figure A.13. Visualization of segmentation results for the wood category on MVTec-AD. The first row represents the input images, with green outlines indicating the ground truth regions. The second row shows the results of anomaly segmentation.

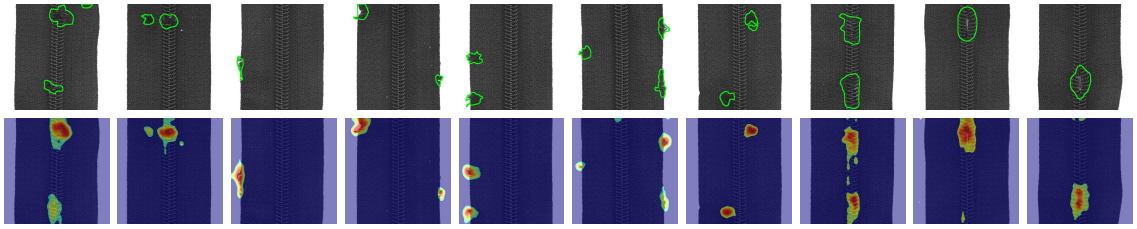


Figure A.14. Visualization of segmentation results for the zipper category on MVTec-AD. The first row represents the input images, with green outlines indicating the ground truth regions. The second row shows the results of anomaly segmentation.

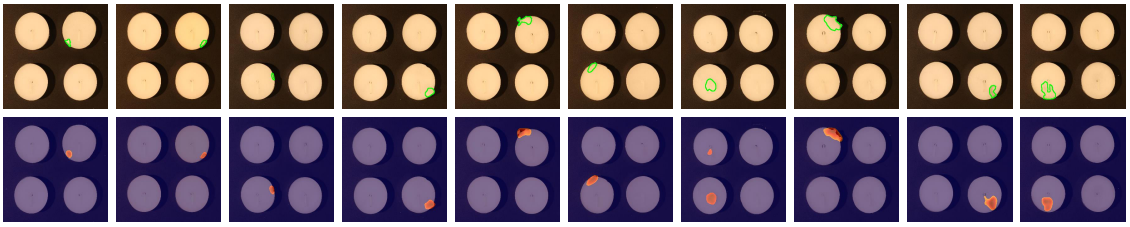


Figure A.15. Visualization of segmentation results for the candle category on VisA. The first row represents the input images, with green outlines indicating the ground truth regions. The second row shows the results of anomaly segmentation.

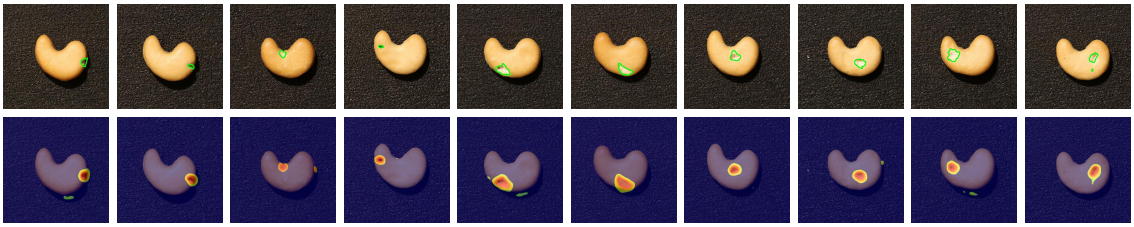


Figure A.16. Visualization of segmentation results for the cashew category on VisA. The first row represents the input images, with green outlines indicating the ground truth regions. The second row shows the results of anomaly segmentation.

Figure A.17. Visualization of segmentation results for the macaroni1 category on VisA. The first row represents the input images, with green outlines indicating the ground truth regions. The second row shows the results of anomaly segmentation.

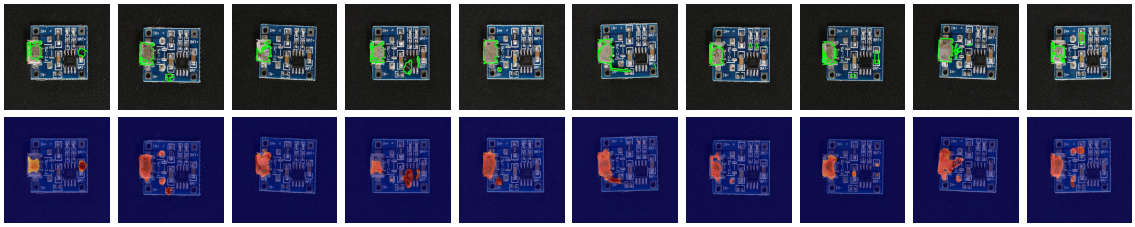


Figure A.18. Visualization of segmentation results for the pcb4 category on VisA. The first row represents the input images, with green outlines indicating the ground truth regions. The second row shows the results of anomaly segmentation.

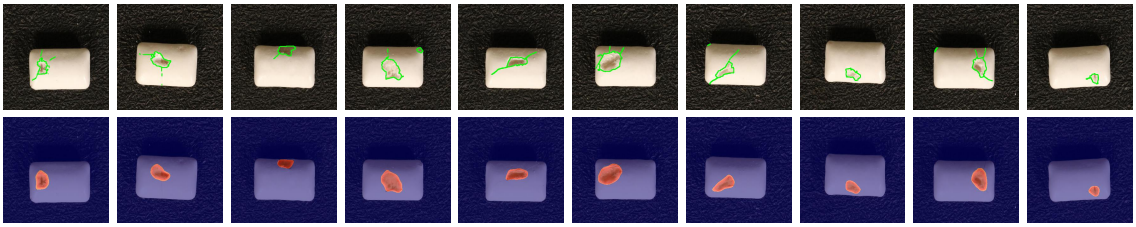


Figure A.19. Visualization of segmentation results for the chewinggum category on VisA. The first row represents the input images, with green outlines indicating the ground truth regions. The second row shows the results of anomaly segmentation.

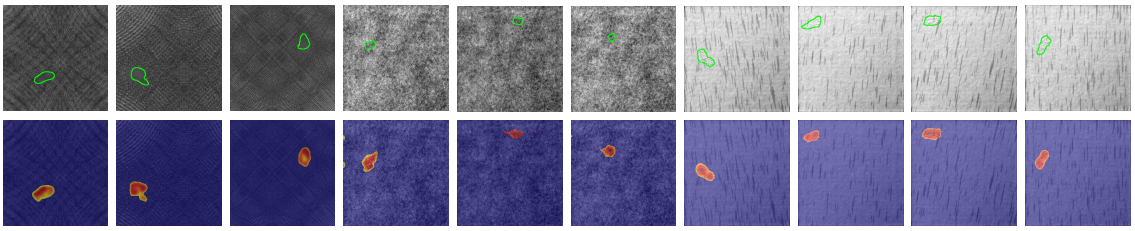


Figure A.20. Visualization of segmentation results on DAGM. The first row represents the input images, with green outlines indicating the ground truth regions. The second row shows the results of anomaly segmentation.

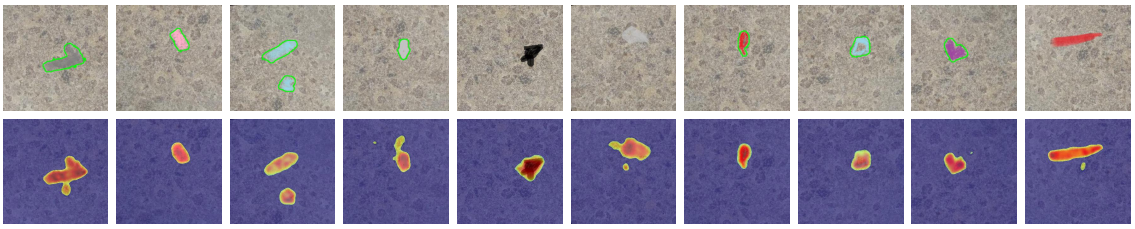


Figure A.21. Visualization of segmentation results for the blotchy category on DTD-Synthetic. The first row represents the input images, with green outlines indicating the ground truth regions. The second row shows the results of anomaly segmentation.

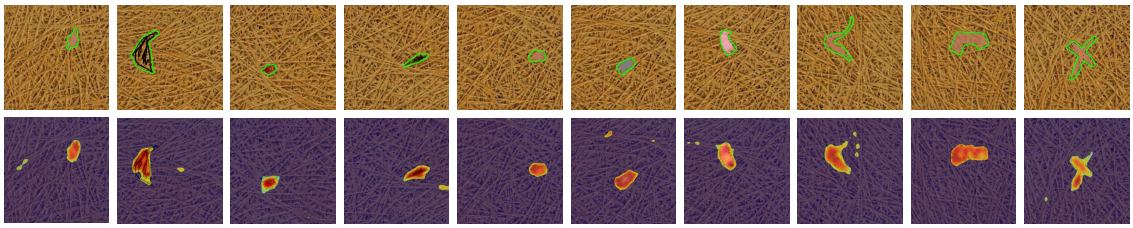


Figure A.22. Visualization of segmentation results for the fibrous category on DTD-Synthetic. The first row represents the input images, with green outlines indicating the ground truth regions. The second row shows the results of anomaly segmentation.

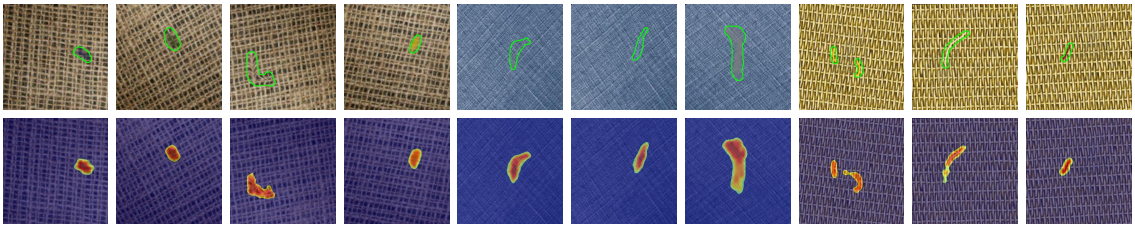


Figure A.23. Visualization of segmentation results for the woven category on DTD-Synthetic. The first row represents the input images, with green outlines indicating the ground truth regions. The second row shows the results of anomaly segmentation.

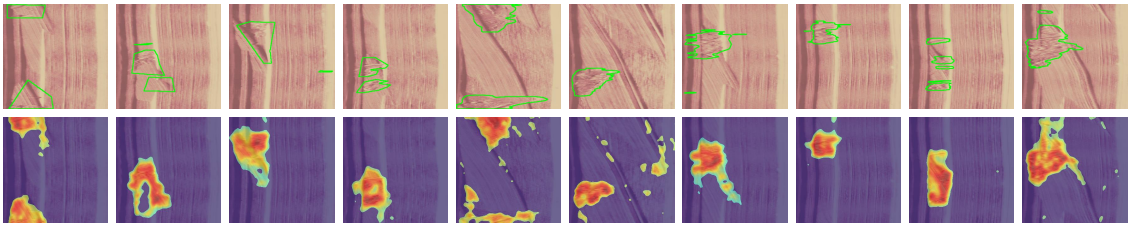


Figure A.24. Visualization of segmentation results for the wood category on BTAD. The first row represents the input images, with green outlines indicating the ground truth regions. The second row shows the results of anomaly segmentation.

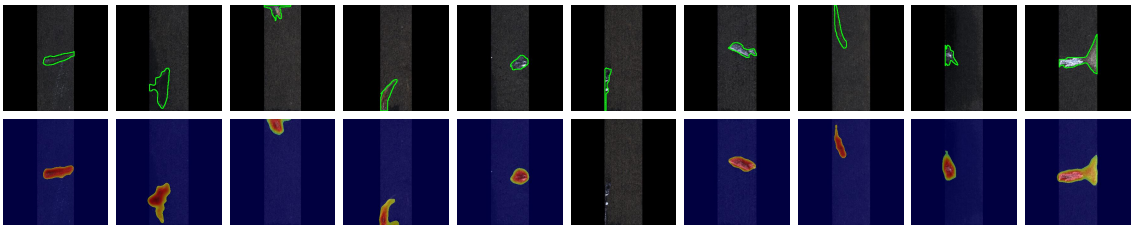


Figure A.25. Visualization of segmentation results on KSDD2. The first row represents the input images, with green outlines indicating the ground truth regions. The second row shows the results of anomaly segmentation.

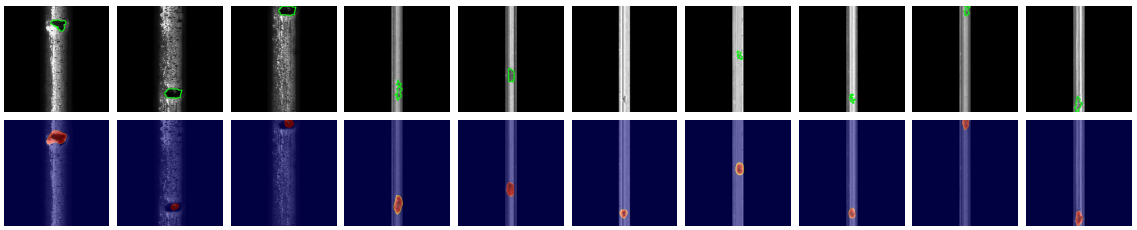


Figure A.26. Visualization of segmentation results on RSDD. The first row represents the input images, with green outlines indicating the ground truth regions. The second row shows the results of anomaly segmentation.

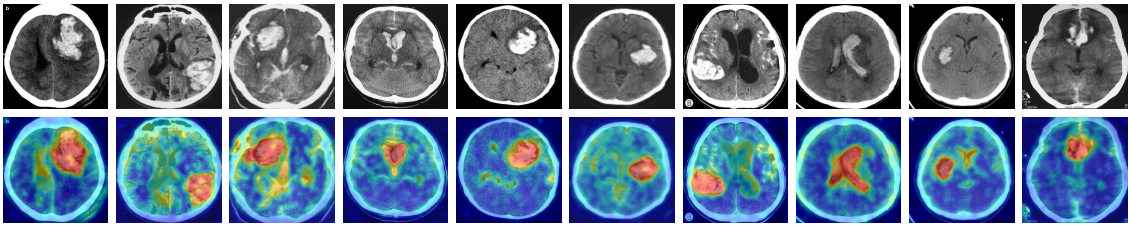


Figure A.27. Visualization of segmentation results on HeadCT. Since there are no pixel-level annotations, this dataset is only used for anomaly classification. The first row represents the input image, while the second row shows the anomaly maps.

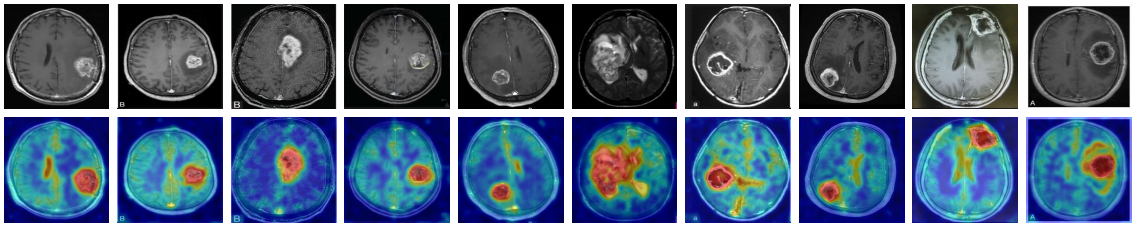


Figure A.28. Visualization of segmentation results on BrainMRI. Since there are no pixel-level annotations, this dataset is only used for anomaly classification. The first row represents the input image, while the second row shows the anomaly maps.

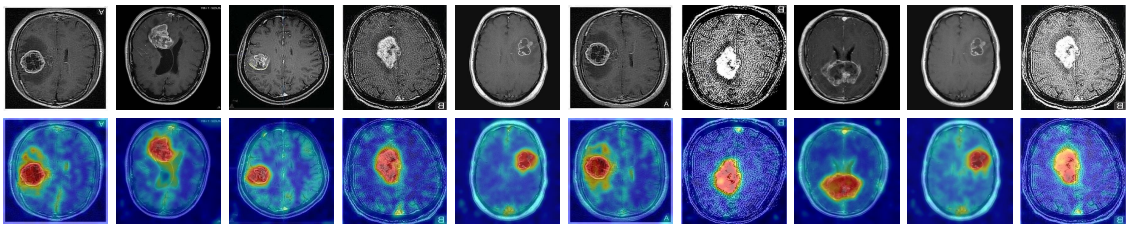


Figure A.29. Visualization of segmentation results on Br35H. Since there are no pixel-level annotations, this dataset is only used for anomaly classification. The first row represents the input image, while the second row shows the anomaly maps.

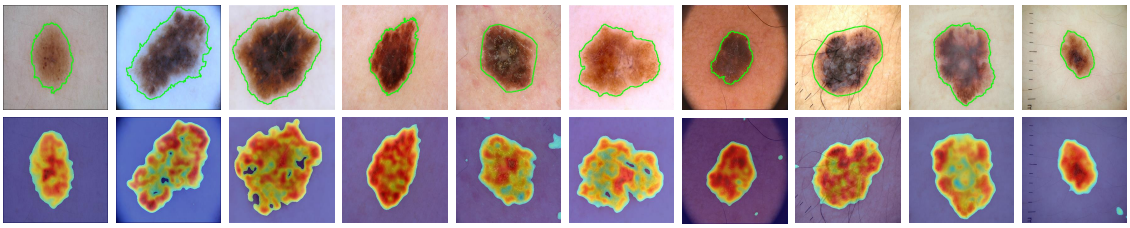


Figure A.30. Visualization of segmentation results on ISIC. The first row represents the input images, with green outlines indicating the ground truth regions. The second row shows the results of anomaly segmentation.

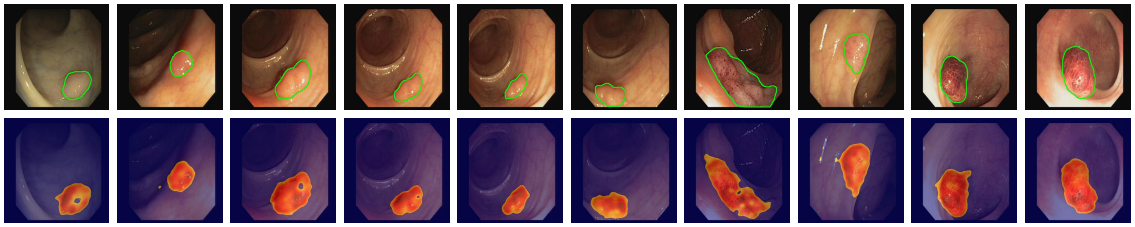


Figure A.31. Visualization of segmentation results on CVC-ClinicDB. The first row represents the input images, with green outlines indicating the ground truth regions. The second row shows the results of anomaly segmentation.

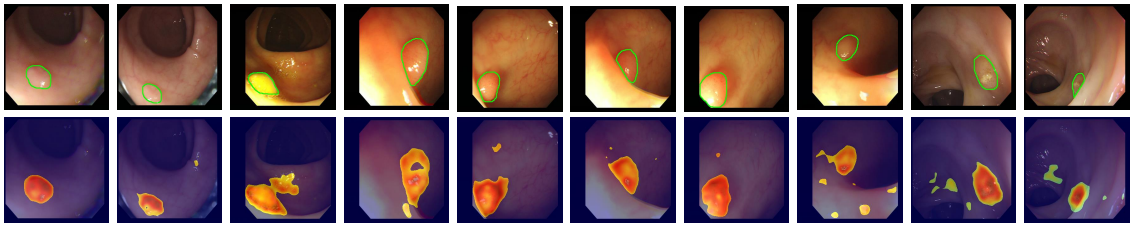


Figure A.32. Visualization of segmentation results on CVC-ColonDB. The first row represents the input images, with green outlines indicating the ground truth regions. The second row shows the results of anomaly segmentation.

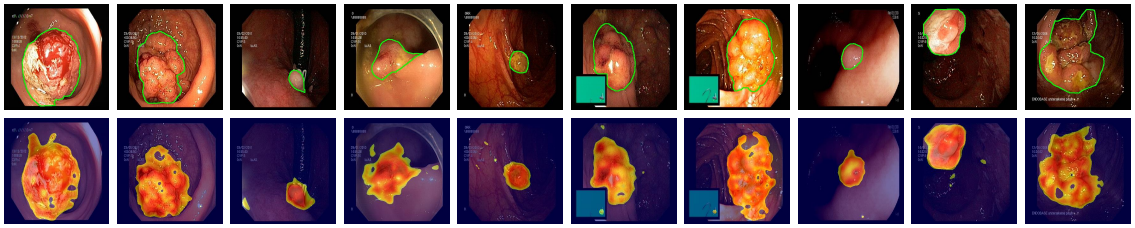


Figure A.33. Visualization of segmentation results on Endo. The first row represents the input images, with green outlines indicating the ground truth regions. The second row shows the results of anomaly segmentation.

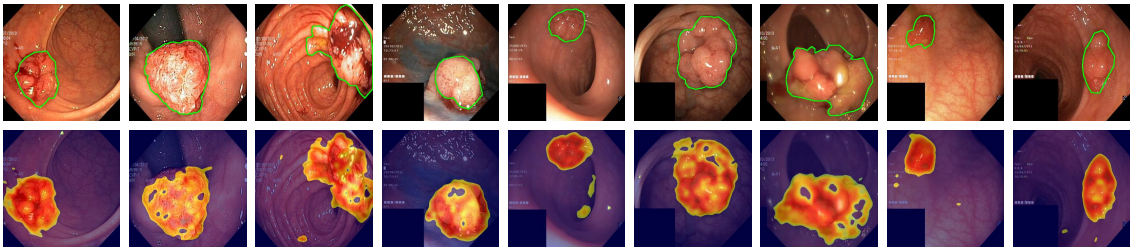


Figure A.34. Visualization of segmentation results on kvasir. The first row represents the input images, with green outlines indicating the ground truth regions. The second row shows the results of anomaly segmentation.

## References


[1] Toshimichi Aota, Lloyd Teh Tzer Tong, and Takayuki Okatani. Zero-shot versus many-shot: Unsupervised texture anomaly detection. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 5564–5572, 2023. 7

[2] 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, 6

[3] Jorge Bernal, F Javier Sánchez, Gloria Fernández-Esparrach, Debora Gil, Cristina Rodríguez, and Fernando Vilariño. Wm-dova maps for accurate polyp highlighting in colonoscopy: Validation vs. saliency maps from physicians. *Computerized medical imaging and graphics*, 43:99–111, 2015. 7

[4] 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. 6

[5] Yunkang Cao, Jiangning Zhang, Luca Frittoli, Yuqi Cheng, Weiming Shen, and Giacomo Boracchi. Adaclip: Adapting clip with hybrid learnable prompts for zero-shot anomaly detection. In *European Conference on Computer Vision*, pages 55–72. Springer, 2025. 1, 3, 5, 7

[6] Xuhai Chen, Yue Han, and Jiangning Zhang. A zero-/few-shot anomaly classification and segmentation method for cvpr 2023 vand workshop challenge tracks 1&2: 1st place on zero-shot ad and 4th place on few-shot ad. *arXiv preprint arXiv:2305.17382*, 2023. 1, 2, 3

[7] Xuhai Chen, Jiangning Zhang, Guanzhong Tian, Haoyang He, Wuhao Zhang, Yabiao Wang, Chengjie Wang, Yunsheng Wu, and Yong Liu. Clip-ad: A language-guided staged dual-path model for zero-shot anomaly detection. *arXiv preprint arXiv:2311.00453*, 2023. 2, 5

[8] Noel CF Codella, David Gutman, M Emre Celebi, Brian Helba, Michael A Marchetti, Stephen W Dusza, Aadi Kalloo, Konstantinos Liopyris, Nabin Mishra, Harald Kittler, et al. Skin lesion analysis toward melanoma detection: A challenge at the 2017 international symposium on biomedical imaging (isbi), hosted by the international skin imaging collaboration (isic). In *2018 IEEE 15th international symposium on biomedical imaging (ISBI 2018)*, pages 168–172. IEEE, 2018. 7

[9] Deutsche Arbeitsgemeinschaft für Mustererkennung. Weakly supervised learning for industrial optical inspection, 2007. 7

[10] A. Hamada. Br35h: Brain tumor detection 2020. *Online. Available: https://www.kaggle.com/datasets/ahmedhamada0/brain-tumor-detection*, 2020. 7

[11] Steven A Hicks, Debesh Jha, Vajira Thambawita, Pål Halvorsen, Hugo L Hammer, and Michael A Riegler. The endotect 2020 challenge: evaluation and comparison of classification, segmentation and inference time for endoscopy. In *Pattern Recognition. ICPR International Workshops and Challenges: Virtual Event, January 10-15, 2021, Proceedings, Part VIII*, pages 263–274. Springer, 2021. 7

[12] Jongheon Jeong, Yang Zou, Taewan Kim, Dongqing Zhang, Avinash Ravichandran, and Onkar Dabeer. Winclip: Zero-/few-shot anomaly classification and segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19606–19616, 2023. 2, 5

[13] Debesh Jha, Pia H Smedsrud, Michael A Riegler, Pål Halvorsen, Thomas De Lange, Dag Johansen, and Håvard D Johansen. Kvasir-seg: A segmented polyp dataset. In *MultiMedia modeling: 26th international conference, MMM 2020, Daejeon, South Korea, January 5–8, 2020, proceedings, part II 26*, pages 451–462. Springer, 2020. 7

[14] Pranita Balaji Kanade and PP Gumaste. Brain tumor detection using mri images. *Brain*, 3(2):146–150, 2015. 7

[15] Diederik P Kingma. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 1

[16] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 2

[17] Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In *2016 fourth international conference on 3D vision (3DV)*, pages 565–571. Ieee, 2016. 2

[18] 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. 6

[19] Zhen Qu, Xian Tao, Mukesh Prasad, Fei Shen, Zhengtao Zhang, Xinyi Gong, and Guiguang Ding. Vcp-clip: A visual context prompting model for zero-shot anomaly segmentation. *arXiv preprint arXiv:2407.12276*, 2024. 7

[20] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 1

[21] T-YLPG Ross and GKHP Dollár. Focal loss for dense object detection. In *proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2980–2988, 2017. 2

[22] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *nature*, 323(6088):533–536, 1986. 1

[23] Mohammadreza Salehi, Niousha Sadjadi, Soroosh Baselizadeh, Mohammad H Rohban, and Hamid R Rabiee. Multiresolution knowledge distillation for anomaly detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14902–14912, 2021. 7

[24] Nima Tajbakhsh, Suryakanth R Gurudu, and Jianming Liang. Automated polyp detection in colonoscopy videos using shape and context information. *IEEE transactions on medical imaging*, 35(2):630–644, 2015. 7

[25] Haomin Yu, Qingyong Li, Yunqiang Tan, Jinrui Gan, Jianzhu Wang, Yangli-ao Geng, and Lei Jia. A coarse-to-fine model for rail surface defect detection. *IEEE Transactions on Instrumentation and Measurement*, 68(3):656–666, 2018. 7

[26] Qihang Zhou, Guansong Pang, Yu Tian, Shibo He, and Jiming Chen. Anomalyclip: Object-agnostic prompt learning for zero-shot anomaly detection. In *The Twelfth International Conference on Learning Representations*, 2023. 1, 3, 5, 7

[27] Yang Zou, Jongheon Jeong, Latha Pemula, Dongqing Zhang, and Onkar Dabeer. Spot-the-difference self-supervised pre-training for anomaly detection and segmentation. In *European Conference on Computer Vision*, pages 392–408. Springer, 2022. 2, 6