Random Word Data Augmentation with CLIP for Zero-Shot Anomaly Detection

Masato Tamura
masato.tamura@hal.hitachi.com

Big Data Analytics Solutions Lab
Hitachi America, Ltd.
2535 Augustine Dr, Santa Clara, California USA

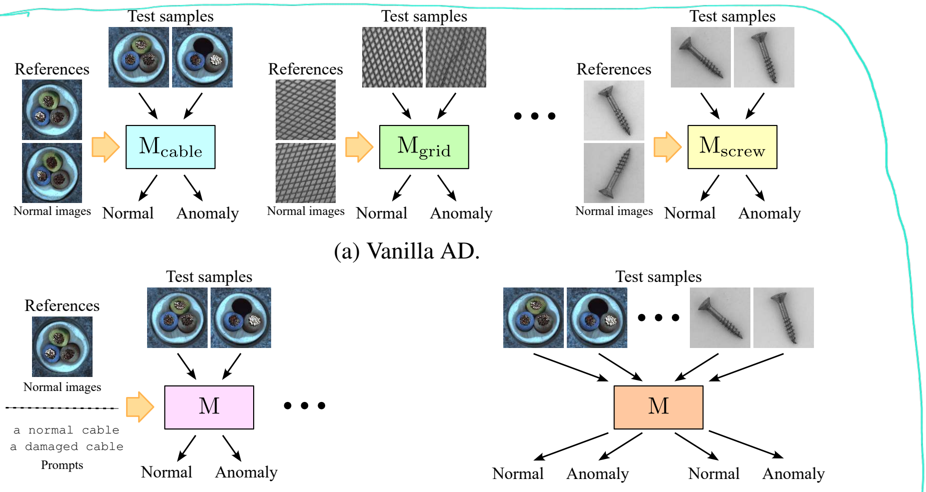
Abstract

This paper presents a novel method that leverages a visual-language model, CLIP, as a data source for zero-shot anomaly detection. Tremendous efforts have been put towards developing anomaly detectors due to their potential industrial applications. Considering the difficulty in acquiring various anomalous samples for training, most existing methods train models with only normal samples and measure discrepancies from the distribution of normal samples during inference, which requires training a model for each object category. The problem of this inefficient training requirement has been tackled by designing a CLIP-based anomaly detector that applies prompt-guided classification to each part of an image in a sliding window manner. However, the method still suffers from the labor of careful prompt ensembling with known object categories. To overcome the issues above, we propose leveraging CLIP as a data source for training. Our method generates text embeddings with the text encoder in CLIP with typical prompts that include words of normal and anomaly. In addition to these words, we insert several randomly generated words into prompts, which enables the encoder to generate a diverse set of normal and anomalous samples. Using the generated embeddings as training data, a feed-forward neural network learns to extract features of normal and anomaly from CLIP's embeddings, and as a result, a category-agnostic anomaly detector can be obtained without any training images. Experimental results demonstrate that our method achieves state-of-the-art performance without laborious prompt ensembling in zero-shot setups.

1 Introduction

Visual anomaly detection (AD) is the task of classifying images as normal or anomaly, where anomalous images typically capture damaged, broken, or defective objects. AD differs from object classification in that anomalous samples are defined as those different from normal samples and thus are not restricted to specific categories. Due to the diverse appearances of anomalous samples, AD remains a challenging problem and thus has been tackled by numerous works [1–5, 7, 8, 11, 14, 15, 19–23, 26, 29, 31–34] for real-world applications.

Considering the fact that a diverse set of anomalous samples is difficult to be obtained, most existing methods [2, 3, 7, 8, 15, 19, 21–23, 26, 29, 31–34] train models with normal samples and then test on both normal and anomalous samples. The distribution of normal samples is modeled via various methods such as one-class classification [22, 32], reconstruction [8, 29], and imitation of anomalous samples [15, 33] during training, and discrepancies



(a) Vanilla AD.

(b) Category-agnostic known-object AD.

(c) Category-agnostic unknown-object AD (ours).

Figure 1: Existing and our AD methods. (a) Existing vanilla AD methods train a model for each object category with normal samples as references and then test the models with the samples of the target object categories. (b) Existing category-agnostic methods do not require category-specific models. However, target object information such as normal images and prompts must be provided during inference. (c) Our method does not need either category-specific models or target object information during inference.

from the distribution are calculated to predict anomaly scores during inference. Since the methods are built on the assumption that normal samples are nearly identical, one model can only be applied to the objects of the same category as depicted in Fig. 1a. This category-specific model requirement demands a significant number of models when the number of target objects is large, which renders the deployment of the methods impractical.

To overcome the aforementioned issue, category-agnostic methods [4, 5, 11, 14, 20] have been proposed. These methods leverage large-scale pre-trained models to extract highly generalizable features and utilize them as references to detect anomalous samples as illustrated in Fig. 1b. In particular, recently proposed WinCLIP [14] demonstrates significant performance improvement over the other category-agnostic methods in a zero-shot AD setup, which extends a highly capable vison-language model, the Contrastive Language-Image Pre-training (CLIP) model [18], to AD and is the most similar to our method.

In this paper, we also concentrate on zero-shot AD with CLIP. However, our method differs from WinCLIP in that we employ CLIP as a data source for training anomaly detectors. We leverage the observation that CLIP's text encoder generates perturbed output embeddings when input sentences are augmented with random words. By this augmentation, we produce a set of highly diverse embeddings with typical prompts containing words of normal and anomaly. These embeddings are then utilized as training data for a feed-forward neural network (FNN), which extracts normal and anomaly features from the embeddings and classifies them accordingly. Since FNN is trained without any object information, our proposed approach is applicable even when target objects are unknown during inference as shown in Fig. 1c. Furthermore, our method does not require any laborious prompt ensembling, which is leveraged in WinCLIP to improve performance.

To summarize, our contributions are three-fold:

- We propose a novel AD method that leverages CLIP as a data source for training an FNN. Since our method does not require any target object information during training or inference, the trained model can be applied to the case where anomalous samples of unknown objects must be detected.
- Our method achieves competitive performance to state-of-the-art methods without any prompt ensembling on two benchmark datasets in challenging zero-shot setups.
- Extensive experiments show the potential use case of the proposed method, where anomalous samples contain objects of ambiguous categories.

2 Related Work

2.1 Anomaly Detection

Most existing AD methods [2, 3, 7, 8, 15, 19, 21–23, 26, 29, 31–34] employ category-specific models to capture the distribution of normal samples during training and then calculate the discrepancies from the distribution during inference. One major trend of training category-specific models is to use unsupervised learning. An early attempt [22] enhances support vector data description, which is a classic algorithm of one-class classification, with deep neural networks. Yi and Yoon [32] extend the method to patch-level classification for precise anomaly segmentation. Gong *et al.* [8] employ another dominant unsupervised approach, where autoencoders are trained to reconstruct normal samples from anomalous ones. Another trend of category-specific models is to leverage self-supervised learning. Golan and Yaniv [7] and Li *et al.* [15] synthesize anomalous samples from normal ones using image transformations and then train models to classify original and synthesized images. Ye *et al.* [31] and Riesta *et al.* [19] propose erasing information in images and forcing models to restore the information during training, which cannot restore anomalous samples perfectly during inference. The aforementioned methods achieve promising results with sophisticated ways of modeling the distribution. However, the inefficiency of category-specific models is non-negligible if the number of categories is significant and thus renders the methods impractical.

To solve the inefficiency of category-specific models, several category-agnostic methods [4, 5, 11, 14, 20] have been proposed, which leverage large-scale pre-trained models. Most of these methods [4, 5, 11, 20] extract features with ImageNet [6] pre-trained models to register the information of normal samples and utilize the information to find anomalous samples. Recently proposed WinCLIP [14] differs from the ImageNet-based methods in that the method employs a vison-language model, CLIP [18], for prompt-guided AD. All of these methods enable models to detect anomalous samples in either zero- and/or few-shot manner due to the generalization capability of the pre-trained models.

Our method also leverages CLIP but differs from WinCLIP in that we use CLIP as a data source for training anomaly detectors. Due to the category-agnostic and highly-diverse samples from CLIP, our method can be applied to samples of unknown object categories and achieve competitive performance without any prompt ensembling in zero-shot AD setups.

2.2 Zero-Shot Image Recognition with CLIP

CLIP [18] is trained with a significant amount of image-text pairs for better generalization performance. During the training, embeddings from the image encoder are forced to have high cosine similarities with the text embeddings of the corresponding pairs from the text encoder. In contrast, the similarities of the embeddings in different pairs are reduced. This contrastive learning enables models to extract highly generalized features from images, and as a result, the models achieve high performance in zero-shot image classification.

Following the success of zero-shot image classification, several image recognition tasks have been tackled with CLIP in a zero-shot manner [9, 24, 30]. Gu *et al.* [9] and Xu *et al.* [30] extend the CLIP-based image classification to object detection and segmentation, respectively. Sato *et al.* [24] tackles zero-shot anomaly action recognition with prompt-guided text embeddings for anomalous behaviors unobserved during training.

We also leverage CLIP but in a different manner; we use CLIP as a data source for training a classifier. The training samples generated by CLIP have high diversity and thus enhance the capability of anomaly detectors without prompt ensembling.

3 Proposed Method

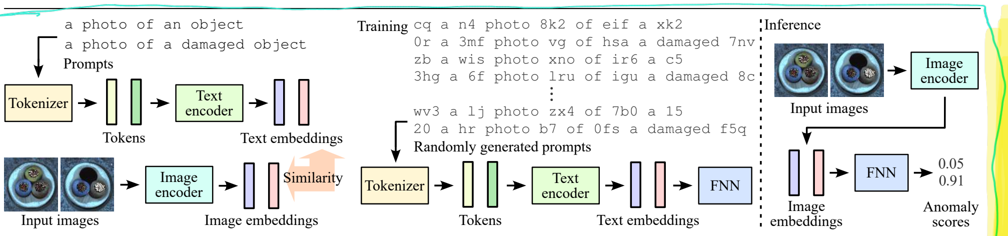
We leverage CLIP [18] as a data source to detect anomaly samples in a zero-shot manner. To elaborate on our method, we first describe prompt-guided AD in Sec. 3.1 and then explain the proposed random word data augmentation in Sec. 3.2, which is the core of our method. Finally, we illustrate the way to train an FNN with samples generated by CLIP in Sec. 3.3. It should be noted that in this section, we explain AD in the case where target object categories are unknown.

3.1 Prompt-Guided Anomaly Detection

Figure 2a shows a prompt-guided AD method. We use two-class prompts as in WinCLIP [14]. However, we do not use state or prompt ensembling, which is utilized in WinCLIP and requires laborious engineering. In prompt-guided AD, two-class prompts, "`a photo of [n] object`" and "`a photo of [a] object`", are first prepared, where words of normal and anomaly are inserted into the locations of "`[n]`" and "`[a]`", respectively. Examples of normal words are "`a`" and "`a normal`", and those of anomlay words are "`a damaged`" and "`a broken`". Two prompts are then transformed into tokens $\boldsymbol{t}^{(n)} \in \mathbb{Z}^{C_t}$ and $\boldsymbol{t}^{(a)} \in \mathbb{Z}^{C_t}$ with the tokenizer in CLIP, where C_t is the maximum token size. The tokens are further transformed into text embeddings $\boldsymbol{e}^{(n,t)} \in \mathbb{R}^{C_e}$ and $\boldsymbol{e}^{(a,t)} \in \mathbb{R}^{C_e}$ as $\boldsymbol{e}^{(n,t)} = f_{tenc}(\boldsymbol{t}^{(n)})$ and $\boldsymbol{e}^{(a,t)} = f_{tenc}(\boldsymbol{t}^{(a)})$, where C_e is the embedding dimension and $f_{tenc}(\cdot)$ is the text encoder in CLIP. The obtained two embeddings are used as guides for AD.

To calculate anomaly scores with the text embeddings, input images must be transformed into embeddings. Given an input image $I \in \mathbb{R}^{3\times H\times W}$, where H and W are the height and width of the input image, an image embedding $\boldsymbol{e}^{(i)} \in \mathbb{R}^{C_e}$ is obtained as $\boldsymbol{e}^{(i)} = f_{ienc}(I)$, where $f_{ienc}(\cdot)$ is the image encoder in CLIP. After normalizing the embeddings as $\bar{\boldsymbol{e}}^{(n,t)} = \frac{\boldsymbol{e}^{(n,t)}}{|\boldsymbol{e}^{(n,t)}|}$, $\bar{\boldsymbol{e}}^{(a,t)} = \frac{\boldsymbol{e}^{(a,t)}}{|\boldsymbol{e}^{(a,t)}|}$, and $\bar{\boldsymbol{e}}^{(i)} = \frac{\boldsymbol{e}^{(i)}}{|\boldsymbol{e}^{(i)}|}$, an anomaly score $s_{pr} \in [0,1]$ is finally obtained using the softmax as $s_{pr} = \frac{\exp(\bar{\boldsymbol{e}}^{(a,t)}\cdot\bar{\boldsymbol{e}}^{(i)}/T)}{\exp(\bar{\boldsymbol{e}}^{(n,t)}\cdot\bar{\boldsymbol{e}}^{(i)}/T)+\exp(\bar{\boldsymbol{e}}^{(a,t)}\cdot\bar{\boldsymbol{e}}^{(i)}/T)}$, where T is a temperature parameter to adjust the sensitivity of the softmax. Following the implementation of CLIP [18], we set the temperature parameter T to 0.01.

Since the prompts do not specify any object categories, this prompt-guided AD can be applied when object categories are unknown. The method can be modified for the known-


a photo of an object
a photo of a damaged object
Prompts
Tokenizer
Tokens
Text encoder
Text embeddings
Input images
Image encoder
Image embeddings
Similarity
Training
cq a n4 photo 8k2 of eif a xk2
0r a 3mf photo vg of hsa a damaged 7nv
zb a wis photo xno of ir6 a c5
3hg a 6f photo lru of igu a damaged 8c
wv3 a 1j photo zx4 of 7b0 a 15
20 a hr photo b7 of 0fs a damaged f5q
Randomly generated prompts
Tokenizer
Tokens
Text encoder
Text embeddings
FNN
Inference
Input images
Image encoder
Image embeddings
FNN
0.05
0.91
Anomaly scores

(a) Prompt-guided AD.

(b) Our method.

Figure 2: Overview of prompt-guided AD and our method.

object case by changing the word “object” in the prompts to a target category name.

3.2 Random Word Data Augmentation

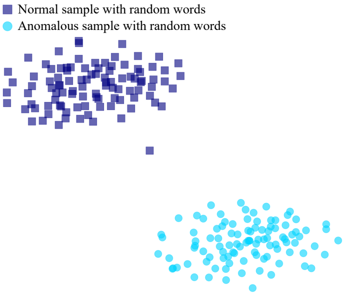


Figure 3: t-SNE plot of generated text embeddings.

To train an FNN-based anomaly detector, we generate training samples by feeding prompts into CLIP with random word data augmentation. Samples are generated based on two prompt templates, “[w$_0$] a [w$_1$] photo [w$_2$] of [w$_3$] [n] [w$_4$]” for normal samples and “[w$_5$] a [w$_6$] photo [w$_7$] of [w$_8$] [a] [w$_9$]” for anomalous samples. At the locations of “[n]” and “[a]”, words of normal and anomaly are inserted, respectively, in the same way as the prompt-guided AD. At the locations of “[w$_i$]”, randomly generated words are inserted.

Words for “[w$_i$]” are generated by randomly selecting letters from the English alphabet and digits. The length of each word is also randomly selected between the range from the minimum length l_{min} to the maximum length l_{max}. The selected letters are concatenated and compose a single word. We insert individually generated words into the prompt templates and finally obtain prompts as depicted in Fig. 2b for generating training samples.

The completed prompts are transformed into text embeddings via the tokenizer and text encoder in CLIP. Suppose we have N_p pairs of completed normal and anomalous prompts, a set of the pairs of tokens $\mathcal{T} = \{(\boldsymbol{t}_i^{(n)}, \boldsymbol{t}_i^{(a)})\}_{i=1}^{N_p}$ is obtained, and then tokens $\boldsymbol{t}_i^{(n)}$ and $\boldsymbol{t}_i^{(a)}$ are transformed into text embeddings as $\boldsymbol{e}_i^{(n)} = f_{tenc}(\boldsymbol{t}_i^{(n)})$ and $\boldsymbol{e}_i^{(a)} = f_{tenc}(\boldsymbol{t}_i^{(a)})$. The obtained set of text embedding pairs $\mathcal{E} = \{(\boldsymbol{e}_i^{(n)}, \boldsymbol{e}_i^{(a)})\}_{i=1}^{N_p}$ is used as a training set for anomaly detectors. We evaluate AD performances with several N_p values, which are reported in Sec. 4.4.

The proposed data augmentation increases the diversity of training samples and thus enhances the performance of anomaly detectors. Figure 3 illustrates the t-SNE [28] plot of the generated text embeddings. It can be seen from the figure that normal and anomalous samples are separately distributed, while samples in either group of normal or anomaly have a certain amount of diversity. Since any object categories are specified in the prompts to generate training samples, the trained anomaly detector can be applied even when target object categories are unknown, resulting in obtaining category-agnostic anomaly detectors.

3.3 Training and Inference

As illustrated in Fig. 2b, we use an FNN as an anomaly detector, which is trained with the generated text embeddings. During training, generated samples $\boldsymbol{e}_i^{(n)}$ and $\boldsymbol{e}_i^{(a)}$ are fed into an FNN, and the binary cross entropy loss is calculated, where normal and anomalous samples are labeled as 0 and 1, respectively. The samples of the same pair are always contained in the same batch because we empirically found that the performance was slightly improved with that strategy. During inference, an input image I is first transformed into an image embedding $\boldsymbol{e}^{(i)}$ as $\boldsymbol{e}^{(i)} = f_{ienc}(I)$, and then an anomaly score s_{FNN} is obtained as $s_{FNN} = \sigma(f_{FNN}(\boldsymbol{e}^{(i)}))$, where $f_{FNN}(\cdot)$ is the trained FNN and $\sigma(\cdot)$ is the sigmoid function. Since the text and image embeddings from CLIP have similar semantics due to the contrastive learning of CLIP, the trained FNN can be applied to the image embeddings even though the FNN is not trained with them. The obtained scores can solely be used to detect anomalous samples or can be combined with any kind of score such as s_{pr} to enhance the performance.

4 Experiments

4.1 Datasets and Evaluation Metrics

To validate the effectiveness of the proposed method, we conduct extensive experiments on three publicly available benchmark datasets: the MVTec-AD [1], VisA [34], and SewerML [10] datasets. The former two datasets are common AD datasets. The MVTec-AD dataset includes 15 object categories, while the VisA dataset comprises 12 object categories. We use the SewerML dataset to evaluate the capability of detecting anomalous samples whose object categories are difficult to define due to their diversities. Although defect types are defined in the dataset, the appearance of each defect is not fixed and thus is challenging to classify, especially in a zero-shot manner. We compare CLIP [18] and our method with this dataset to validate the effectiveness when ambiguous anomalous samples must be detected. Only the validation set of the SewerML dataset is used for this evaluation.

Following the work of WinCLIP [14], we report the evaluation metrics of Area Under the Receiver Operating Characteristic (AUROC), Area Under the Precision-Recall curve (AUPR), and F_1-score at optimal threshold (F_1-max). The reported values of our method are the mean and standard deviation of trials with 10 random seeds.

4.2 Implementation Details

In all the experiments, we use OpenCLIP[1] as the implementation of CLIP [18] and the LAION-400M [25]-based CLIP with ViT-B/16+ [12], which is also used by the work of WinCLIP [14], for fair comparisons. The FNN has four linear layers with batch normalization [13], ReLU activations, and dropout [27]. The network is trained for 2 epochs using the AdamW [17] optimizer with a batch size of 128, the initial learning rate of 10^{-3} and the weight decay of 10^{-4}. The learning rate is decayed after 1 epoch. Note that each batch comprises paired normal and anomalous samples as described in Sec. 3.3. Unless otherwise noted, we use multi-crop data augmentation at test time for both CLIP and our method because WinCLIP also uses the multi-crop strategy.

[1] https://github.com/mlfoundations/open_clip

Table 1: Comparison against state-of-the-art methods.

		MVTec-AD			VisA		
Setup	Method	AUROC	AUPR	F_1-max	AUROC	AUPR	F_1-max
0-shot (Object unknown)	CLIP [18]	91.5	95.7	92.0	76.5	80.5	78.1
	Ours	91.0±0.4	95.4±0.3	92.2±0.3	78.1±1.1	81.3±0.9	79.8±0.4
	CLIP + ours	**92.2±0.3**	**96.0±0.2**	**92.8±0.2**	**78.2±0.8**	**81.5±0.8**	**79.9±0.3**
0-shot (Object known)	WinCLIP [14]	91.8	**96.5**	92.9	78.1	81.2	79.0
	CLIP [18]	92.6	96.3	93.0	76.3	80.4	78.8
	CLIP + ours	**93.0±0.3**	96.4±0.2	**93.1±0.2**	**79.8±0.6**	**82.8±0.6**	**79.9±0.2**
1-shot	SPADE [4]	81.0±2.0	90.6±0.8	90.3±0.8	79.5±4.0	82.0±3.3	80.7±1.9
	PaDiM [5]	76.6±3.1	88.1±1.7	88.2±1.1	62.8±5.4	68.3±4.0	75.3±1.2
	PatchCore [20]	83.4±3.0	92.2±1.5	90.5±1.5	79.9±2.9	82.8±2.3	81.7±1.6
	WinCLIP [14]	93.1±2.0	96.5±0.9	93.7±1.1	**83.8±4.0**	85.1±4.0	83.1±1.7
	CLIP + ours	**93.3±0.5**	**96.7±0.2**	**94.0±0.3**	83.4±1.7	**85.8±1.8**	**83.6±0.8**
2-shot	SPADE [4]	82.9±2.6	91.7±1.2	91.1±1.0	80.7±5.0	82.3±4.3	81.7±2.5
	PaDiM [5]	78.9±3.1	89.3±1.7	89.2±1.1	67.4±5.1	71.6±3.8	75.7±1.8
	PatchCore [20]	86.3±3.3	93.8±1.7	92.0±1.5	81.6±4.0	84.8±3.2	82.5±1.8
	RegAD [11]	85.7	–	–	–	–	–
	WinCLIP [14]	**94.4±1.3**	**97.0±0.7**	**94.4±0.8**	84.6±2.4	85.8±2.7	83.0±1.4
	CLIP + ours	94.0±0.7	96.9±0.3	94.1±0.3	**85.6±1.4**	**87.5±1.6**	**84.1±1.0**
4-shot	SPADE [4]	84.8±2.5	92.5±1.2	91.5±0.9	81.7±3.4	83.4±2.7	82.1±2.1
	PaDiM [5]	80.4±2.5	90.5±1.6	90.2±1.2	72.8±2.9	75.6±2.2	78.0±1.2
	PatchCore [20]	88.8±2.6	94.5±1.5	92.6±1.6	85.3±2.1	87.5±2.1	84.3±1.3
	RegAD [11]	88.2	–	–	–	–	–
	WinCLIP [14]	**95.2±1.3**	**97.3±0.6**	**94.7±0.8**	**87.3±1.8**	**88.8±1.8**	84.2±1.6
	CLIP + ours	94.5±0.7	97.1±0.3	94.4±0.3	86.6±0.9	88.4±1.3	**84.5±0.6**

For the random word data augmentation, the minimum word length l_{min} is set to 5, and the maximum word length l_{max} is set to 10. By default, 10,000 pairs of normal and anomalous samples are generated with "a" for a word of normal and "a damaged" for a word of anomaly. The performances with other settings are analyzed in Sec. 4.4.

As denoted in Sec. 3.3, the scores of the FNN can be combined with other scores. In zero-shot setups, the performances with $s_{pr} + s_{FNN}$ (indicated by "CLIP + ours") as well as with the individual score s_{FNN} (indicated by "ours") are reported. For future reference, we also report the performances of few-shot setups. In these setups, an additional score s_{img} is calculated based on the similarities between the image embeddings of test samples and reference normal samples as described in the paper of WinCLIP. The performances are evaluated with the added score $s_{pr} + s_{FNN} + s_{img}$ (indicated by "CLIP + ours").

4.3 Comparison against State of The Art

We compare our method against state-of-the-art zero/few-shot AD methods. Table 1 shows the comparison results. For zero-shot AD, we evaluate performances in two setups. One is an unknown-object setup, where we assume that target object categories are unknown and thus a word "object" is used in CLIP's prompts. The other is known-object setup, where we assume that target object categories are known and thus inserted into CLIP's prompts. As can be seen in the table, our method outperforms CLIP's [18] prompt-guided AD and WinCLIP [14] with almost all the evaluation metrics when combined with CLIP. In partic-

Table 2: Comparison against state-of-the-art methods without multiple crops on the MVTec-AD dataset in the zero-shot known-object setup.

Method	Prompt ens.	AUROC	AUPR	F_1-max
WinCLIP [14]	✓	90.8	96.1	**92.5**
CLIP [18]		89.8	95.4	92.1
Ours		89.6±0.6	95.5±0.2	91.5±0.3
CLIP + ours		**91.0±0.3**	**96.2±0.2**	**92.5±0.2**

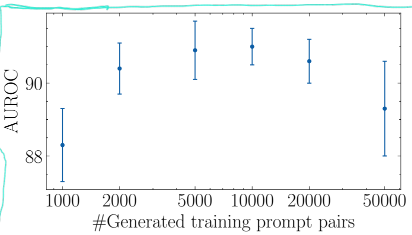


Figure 4: AUROCs with the various numbers of training prompt pairs in the zero-shot unknown-object setup.

Table 3: AUROCs with various word pairs for normal and anomaly in the zero-shot unknown-object setup. The left value in each cell is the result of CLIP [18], and the right value is that of CLIP + ours.

	"a damaged"	"a broken"	"a defective"	"an anomalous"
"an"	91.5/**92.2±0.3**	**87.5**/86.1±0.6	79.4/**85.7±0.5**	67.6/**73.7±0.3**
"a normal"	89.3/**90.5±0.2**	87.3/**88.6±0.5**	81.8/**84.5±0.9**	69.1/**71.9±1.0**
"a good"	88.4/**89.6±0.3**	86.1/**87.0±0.5**	80.6/**86.3±0.4**	68.6/**73.0±0.9**
"a flawless"	88.5/**90.3±0.4**	85.7/**86.0±0.8**	77.7/**84.5±0.6**	68.8/**75.8±0.5**

ular, the performance gaps between the existing and our methods are larger in the zero-shot unknown-object setup than in the known-object setup, which indicates the effectiveness of detecting ambiguous anomalous samples.

The performances of the few-shot setups show that our method achieves competitive performances even when some normal reference images are provided. These results suggest that our method can be combined with state-of-the-art methods and enhance their performance.

We also compare the performances of CLIP [18], WinCLIP [14], and our method without the multi-crop strategy. If the multi-crop is eliminated from WinCLIP, the performance purely depends on the prompt ensembling, which is a suitable baseline to evaluate the capability of our method. Table 2 shows the comparison results. As shown in the table, our method achieves better performance than WinCLIP. The results indicate that our random word data augmentation can improve performance better than prompt ensembling.

4.4 Analysis on Data Augmention

We first analyze the effect of the number of prompt pairs N_p for training the FNN. Figure 4 shows the performances on the MVTec-AD dataset with regard to the different values of N_p in the unknown-object setup. As shown in the figure, the best performance is achieved when N_p is 10,000, and fewer or more pairs degrade the performance. The lower performance with fewer samples is attributed to insufficient training data, and that with more samples is probably due to over-fitting. As analyzed in Sec. 4.5, embeddings generated from our random data augmentation have a different distribution from that of the embeddings generated from natural sentences. A large number of such training samples force models to have an incorrect decision boundary and thus degrades the performance.

We then analyze the effect of normal and anomaly word choices in guiding prompts and prompt templates. Table 3 shows the evaluation results of CLIP/CLIP + ours. Each row and column shows the performances with the indicated word as normal and anomaly in the prompts. As illustrated in the table, our method improves the performances of CLIP

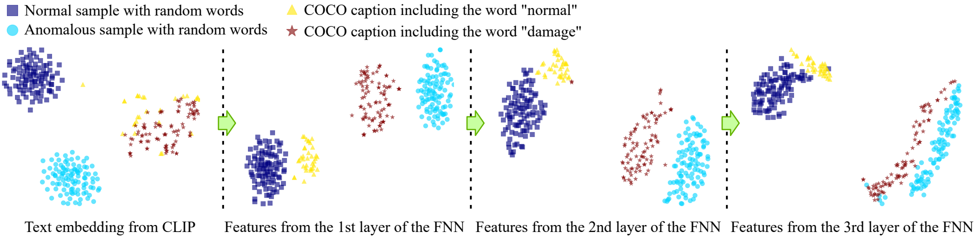


Figure 5: t-SNE plots of feature embeddings from different layers.

Table 4: Comparison of CLIP and our method on the SewerML dataset.

Method	AUROC	AUPR	F_1-max
CLIP [18]	79.1	77.7	75.2
Ours	**82.6±1.3**	**81.0±1.3**	**77.4±0.9**
CLIP + Ours	81.7±0.9	80.2±0.7	76.9±0.8

with any word pairs except for the pair of "an" and "a broken". In particular, our method significantly improves performances in the case where word pairs are inappropriate for CLIP. The results suggest that our method can show word-agnostic performance improvement and can boost the robustness of the prompt-guided AD.

4.5 Analysis on Feature Embeddings

To analyze how the FNN learns to classify normal and anomalous samples with our random word data augmentation, we obtain natural sentences that contain a word of either "normal" or "damage" from COCO captions [16] and plot the embeddings of those natural sentences in addition to the augmented prompts with t-SNE [28]. Figure 5 shows the plotted results. As shown in the figure, CLIP's embeddings of the natural sentences and augmented prompts have different domains. However, the embeddings from the FNN layers are gradually split by their sample types rather than their domains. These plots indicate that the trained FNN learns to extract features containing information on normality and thus successfully classifies anomalous samples of arbitrary object categories.

4.6 Potential Application

To validate the effectiveness of our method in detecting ambiguous anomalous samples, we evaluate the performances of CLIP [18] and our method on the SewerML dataset [10]. Table 4 shows the comparison results. As shown in the table, our method without the prompt-guided AD achieves the best performance among the three methods. The reason for the best performance without the prompt-guided AD is probably because CLIP cannot handle the diversity of defects in the dataset and thus degrades the performance rather than improving it. In contrast, our FNN is trained with a diverse set of normal and anomalous samples and thus can achieve better performance.

5 Conclusion

We propose a novel zero-shot category-agnostic AD method that leverages CLIP as a data source for training anomaly detectors. Our method differs from existing methods in that our method does not include object categories in prompts to generate training samples and thus can be applied to the case where object categories are unknown. Furthermore, training samples generated by our method are so diverse that our method can achieve significant performance even without prompt ensembling. We perform extensive experiments and show the effectiveness of our method.

References

[1] Paul Bergmann, Michael Fauser, David Sattlegger, and Carsten Steger. MVTec AD – A comprehensive real-world dataset for unsupervised anomaly detection. In *CVPR*, 2019.

[2] Paul Bergmann, Michael Fauser, David Sattlegger, and Carsten Steger. Uninformed students: Student-teacher anomaly detection with discriminative latent embeddings. In *CVPR*, 2020.

[3] Paul Bergmann, Kilian Batzner, Michael Fauser, David Sattlegger, and Carsten Steger. Beyond dents and scratches: Logical constraints in unsupervised anomaly detection and localization. *International Journal of Computer Vision*, 130(4):947–969, 2022.

[4] Niv Cohen and Yedid Hoshen. Sub-image anomaly detection with deep pyramid correspondences, 2020. arXiv:2005.02357.

[5] Thomas Defard, Aleksandr Setkov, Angelique Loesch, and Romaric Audigier. PaDiM: A patch distribution modeling framework for anomaly detection and localization. In *ICPRW*, 2021.

[6] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *CVPR*, 2009.

[7] Izhak Golan and Ran El-Yaniv. Deep anomaly detection using geometric transformations. In *NeurIPS*, 2018.

[8] Dong Gong, Lingqiao Liu, Vuong Le, Budhaditya Saha, Moussa Reda Mansour, Svetha Venkatesh, and Anton van den Hengel. Memorizing normality to detect anomaly: Memory-augmented deep autoencoder for unsupervised anomaly detection. In *ICCV*, 2019.

[9] Xiuye Gu, Tsung-Yi Lin, Weicheng Kuo, and Yin Cui. Open-vocabulary object detection via vision and language knowledge distillation. In *ICLR*, 2022.

[10] Joakim Bruslund Haurum and Thomas B. Moeslund. Sewer-ML: A multi-label sewer defect classification dataset and benchmark. In *CVPR*, 2021.

[11] Chaoqin Huang, Haoyan Guan, Aofan Jiang, Ya Zhang, Michael Spratlin, and Yanfeng Wang. Registration based few-shot anomaly detection. In *ECCV*, 2022.

[12] Gabriel Ilharco, Mitchell Wortsman, Ross Wightman, Cade Gordon, Nicholas Carlini, Rohan Taori, Achal Dave, Vaishaal Shankar, Hongseok Namkoong, John Miller, Hannaneh Hajishirzi, Ali Farhadi, and Ludwig Schmidt. OpenCLIP, 2021.

[13] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, 2015.

[14] Jongheon Jeong, Yang Zou, Taewan Kim, Dongqing Zhang, Avinash Ravichandran, and Onkar Dabeer. WinCLIP: Zero-/few-shot anomaly classification and segmentation. In *CVPR*, 2023.

[15] Chun-Liang Li, Jinsung Yoon, Kihyuk Sohn, and Tomas Pfister. CutPaste: Self-supervised learning for anomaly detection and localization. In *CVPR*, 2021.

[16] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: Common objects in context. In *ECCV*, 2014.

[17] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2019.

[18] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *ICML*, 2021.

[19] Nicolae-Catalin Ristea, Neelu Madan, Radu Tudor Ionescu, Kamal Nasrollahi, Fahad Shahbaz Khan, Thomas B Moeslund, and Mubarak Shah. Self-supervised predictive convolutional attentive block for anomaly detection. In *CVPR*, 2022.

[20] Karsten Roth, Latha Pemula, Joaquin Zepeda, Bernhard Schölkopf, Thomas Brox, and Peter Gehler. Towards total recall in industrial anomaly detection. In *CVPR*, 2022.

[21] Marco Rudolph, Bastian Wandt, and Bodo Rosenhahn. Same same but DifferNet: Semi-supervised defect detection with normalizing flows. In *WACV*, 2021.

[22] Lukas Ruff, Robert Vandermeulen, Nico Goernitz, Lucas Deecke, Shoaib Ahmed Siddiqui, Alexander Binder, Emmanuel Müller, and Marius Kloft. Deep one-class classification. In *ICML*, 2018.

[23] Mohammadreza Salehi, Niousha Sadjadi, Soroosh Baselizadeh, Mohammad Hossein Rohban, and Hamid R. Rabiee. Multiresolution knowledge distillation for anomaly detection. In *CVPR*, 2021.

[24] Fumiaki Sato, Ryo Hachiuma, and Taiki Sekii. Prompt-guided zero-shot anomaly action recognition using pretrained deep skeleton features. In *CVPR*, 2023.

[25] 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. 2021. arXiv:2111.02114.

[26] Shelly Sheynin, Sagie Benaim, and Lior Wolf. A hierarchical transformation-discriminating generative model for few shot anomaly detection. In *ICCV*, 2021.

[27] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958, 2014.

[28] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605, 2008.

[29] Jhih-Ciang Wu, Ding-Jie Chen, Chiou-Shann Fuh, and Tyng-Luh Liu. Learning unsupervised metaformer for anomaly detection. In *ICCV*, 2021.

[30] Mengde Xu, Zheng Zhang, Fangyun Wei, Yutong Lin, Yue Cao, Han Hu, and Xiang Bai. A simple baseline for open-vocabulary semantic segmentation with pre-trained vision-language model. In *ECCV*, 2022.

[31] Fei Ye, Chaoqin Huang, Jinkun Cao, Maosen Li, Ya Zhang, and Cewu Lu. Attribute restoration framework for anomaly detection. *IEEE Transactions on Multimedia*, 24: 116–127, 2019.

[32] Jihun Yi and Sungroh Yoon. Patch SVDD: Patch-level SVDD for anomaly detection and segmentation. In *ACCV*, 2021.

[33] Vitjan Zavrtanik, Matej Kristan, and Danijel Skocaj. DRAEM – A discriminatively trained reconstruction embedding for surface anomaly detection. In *ICCV*, 2021.

[34] Yang Zou, Jongheon Jeong, Latha Pemula, Dongqing Zhang, and Onkar Dabeer. SPot-the-Difference self-supervised pre-training for anomaly detection and segmentation. In *ECCV*, 2022.

A Detailed Quantitative Results

We report category-wise performance of CLIP [18], WinCLIP [14], and ours for future reference. Our performances are those with the score combination of CLIP and our trained feed-forward network. The reported values of our method are the mean and standard deviation of trials with 10 random seeds.

Table 5: Class-wise AUROC on the MVTec-AD dataset.

	0-shot (Object unknown)		0-shot (Object known)		1-shot		2-shot		4-shot	
Category	CLIP	CLIP + ours	WinCLIP	CLIP + ours	WinCLIP	CLIP + ours	WinCLIP	CLIP + ours	WinCLIP	CLIP + ours
Bottle	99.4	97.1±1.1	99.2	97.0±0.8	98.2±0.9	97.8±0.6	99.3±0.3	98.1±0.7	99.3±0.4	98.2±0.6
Cable	83.0	84.2±0.6	86.5	86.4±1.4	88.9±1.9	90.8±1.1	88.4±0.7	91.3±2.0	90.9±0.9	92.1±0.8
Capsule	85.3	87.3±2.0	72.9	88.7±2.8	72.3±6.8	76.8±6.4	77.3±8.8	83.6±7.1	82.3±8.9	87.8±7.9
Carpet	100	100±0.0	100	100±0.0	99.8±0.3	100±0.0	99.8±0.3	100±0.0	100±0.0	100±0.0
Grid	99.2	99.0±0.4	98.8	98.7±0.2	99.5±0.3	99.2±0.5	99.4±0.2	99.1±0.4	99.6±0.1	99.6±0.2
Hazelnut	92.0	94.3±1.1	93.9	94.0±0.5	97.5±1.4	94.9±0.6	98.3±0.7	95.0±0.5	98.4±0.4	95.1±0.5
Leather	100	100±0.0	100	100±0.0	99.9±0.0	100±0.0	99.9±0.0	99.9±0.0	100±0.0	100±0.0
Metal nut	94.4	96.1±1.3	97.1	94.7±1.7	98.7±0.8	96.1±1.3	99.4±0.2	96.1±1.4	99.5±0.2	96.3±1.3
Pill	88.3	88.6±1.6	79.1	90.2±1.0	91.2±2.1	92.5±1.3	92.3±0.7	92.6±1.2	92.8±1.0	92.6±1.3
Screw	76.1	76.6±1.5	83.3	75.5±2.1	86.4±0.9	77.4±1.8	86.0±2.1	77.5±2.1	87.9±1.2	77.6±2.1
Tile	99.4	99.5±0.2	100	99.5±0.2	99.9±0.0	99.6±0.1	99.9±0.2	99.6±0.1	99.9±0.1	99.6±0.1
Toothbrush	92.8	88.4±3.5	87.5	94.0±0.9	92.2±4.9	94.7±1.3	97.5±1.6	95.0±1.0	96.7±2.6	96.4±1.5
Transistor	79.7	86.6±1.4	88.0	88.9±1.3	83.4±3.8	89.4±1.3	85.3±1.7	89.6±1.4	85.7±2.5	89.6±1.3
Wood	97.8	97.6±0.6	99.4	98.1±0.5	99.9±0.1	98.7±0.6	99.9±0.1	98.8±0.5	99.8±0.3	98.8±0.6
Zipper	85.5	87.4±1.0	91.5	88.8±0.1	88.8±5.9	91.6±2.9	94.0±1.4	93.8±1.0	94.5±0.5	94.3±1.0

Table 6: Class-wise AUPR on the MVTec-AD dataset.

	0-shot (Object unknown)		0-shot (Object known)		1-shot		2-shot		4-shot	
Category	CLIP	CLIP + ours	WinCLIP	CLIP + ours	WinCLIP	CLIP + ours	WinCLIP	CLIP + ours	WinCLIP	CLIP + ours
Bottle	99.8	99.2±0.3	99.8	99.2±0.2	99.4±0.3	99.4±0.2	99.8±0.1	99.4±0.2	99.8±0.1	99.5±0.2
Cable	88.3	89.3±0.5	91.2	91.6±1.0	93.2±1.1	94.6±0.8	92.9±0.6	94.9±1.2	94.4±0.3	95.4±0.5
Capsule	96.8	96.8±0.7	91.5	97.0±0.9	91.6±2.7	93.4±2.2	93.3±3.6	95.6±2.2	95.1±3.3	96.6±2.6
Carpet	100	100±0.0	100	100±0.0	99.9±0.1	100±0.0	99.9±0.1	100±0.0	100±0.0	100±0.0
Grid	99.7	99.7±0.1	99.6	99.6±0.0	99.9±0.1	99.7±0.2	99.8±0.1	99.7±0.1	99.9±0.0	99.9±0.1
Hazelnut	96.0	96.7±0.6	96.9	96.2±0.4	98.6±0.7	96.8±0.5	99.1±0.4	96.8±0.4	99.1±0.2	96.9±0.5
Leather	100	100±0.0	100	100±0.0	100±0.0	100±0.0	100±0.0	100±0.0	100±0.0	100±0.0
Metal nut	98.8	99.1±0.3	99.3	98.8±0.4	99.7±0.2	99.1±0.3	99.9±0.0	99.1±0.3	99.9±0.1	99.2±0.3
Pill	97.5	97.6±0.4	95.7	98.0±0.2	98.3±0.5	98.5±0.3	98.6±0.1	98.5±0.3	98.6±0.2	98.5±0.3
Screw	90.3	90.7±0.8	93.1	90.1±1.2	94.2±0.6	91.0±1.1	94.1±1.5	91.1±1.1	94.9±0.8	91.1±1.1
Tile	99.8	99.8±0.1	100	99.8±0.1	100±0.0	99.8±0.1	100±0.1	99.8±0.0	100±0.0	99.8±0.1
Toothbrush	97.7	95.4±1.7	95.6	97.6±0.4	96.7±2.0	97.9±0.6	99.0±0.6	98.0±0.4	98.7±1.1	98.6±0.6
Transistor	76.5	80.4±1.7	87.1	82.3±2.1	79.0±4.0	83.3±2.1	80.7±2.3	83.5±2.2	80.7±3.2	83.6±2.1
Wood	99.3	99.3±0.2	99.8	99.4±0.2	100±0.0	99.6±0.2	100±0.0	99.6±0.2	99.9±0.1	99.6±0.2
Zipper	95.6	96.1±0.3	97.5	96.7±0.0	96.8±1.8	97.5±0.9	98.3±0.4	98.1±0.4	98.5±0.2	98.3±0.3

Table 7: Class-wise F_1-max on the MVTec-AD dataset.

	0-shot (Object unknown)		0-shot (Object known)		1-shot		2-shot		4-shot	
Category	CLIP	CLIP + ours	WinCLIP	CLIP + ours	WinCLIP	CLIP + ours	WinCLIP	CLIP + ours	WinCLIP	CLIP + ours
Bottle	98.4	95.2±1.1	97.6	95.9±0.7	96.5±1.3	96.5±0.6	97.7±0.7	96.6±0.7	97.8±0.6	96.6±0.7
Cable	84.5	85.7±0.8	84.5	84.4±1.2	86.1±1.3	87.7±1.0	85.2±0.7	87.6±2.0	87.2±0.6	87.8±1.0
Capsule	90.7	92.6±0.8	91.4	93.3±0.8	91.6±0.7	93.1±0.7	92.1±0.7	94.0±0.8	92.5±0.5	94.7±0.8
Carpet	99.4	99.6±0.4	99.4	99.6±0.4	99.2±0.8	100±0.0	99.3±0.7	100±0.0	99.9±0.2	100±0.0
Grid	97.3	97.8±0.6	98.2	97.3±0.0	98.9±0.4	98.1±0.7	99.1±0.0	98.1±0.5	99.1±0.0	98.4±0.4
Hazelnut	88.9	91.9±1.7	89.7	91.8±1.0	94.7±2.3	92.7±0.9	95.6±1.6	92.9±0.8	96.2±1.0	93.0±0.8
Leather	100	100±0.0	100	100±0.0	99.5±0.0	99.7±0.3	99.7±0.2	99.5±0.3	99.8±0.2	99.9±0.2
Metal nut	93.3	95.0±1.1	96.3	94.0±1.2	97.7±1.0	95.0±1.0	98.4±0.5	94.9±1.2	98.5±0.6	95.1±1.0
Pill	93.7	93.4±0.6	91.6	93.9±0.3	93.8±0.7	95.1±0.7	94.3±0.4	95.2±0.7	94.1±0.4	95.1±0.8
Screw	86.9	86.9±0.0	87.4	86.9±0.2	88.5±0.3	86.9±0.3	89.0±0.6	87.0±0.4	89.6±0.7	87.0±0.4
Tile	97.7	97.8±0.6	99.4	97.8±0.6	98.9±0.2	97.9±0.5	99.2±0.3	97.9±0.5	99.2±0.3	98.0±0.5
Toothbrush	93.1	90.2±0.9	87.9	92.7±0.7	94.1±1.9	94.1±1.1	96.7±1.8	93.7±0.9	96.8±2.3	95.3±1.6
Transistor	69.7	79.6±1.8	79.5	81.7±1.5	75.1±3.1	81.6±1.7	75.9±2.4	81.7±1.7	76.6±2.8	81.8±1.5
Wood	95.7	95.8±0.6	98.3	96.0±0.4	99.4±0.3	97.2±0.6	99.5±0.4	97.2±0.6	99.2±0.9	97.1±0.7
Zipper	90.0	91.2±0.8	92.9	91.9±0.1	92.1±2.5	93.7±2.1	94.4±0.3	95.5±0.3	94.7±0.4	95.5±0.3

Table 8: Class-wise AUROC on the VisA dataset.

	0-shot (Object unknown)		0-shot (Object known)		1-shot		2-shot		4-shot	
Category	CLIP	CLIP + ours	WinCLIP	CLIP + ours	WinCLIP	CLIP + ours	WinCLIP	CLIP + ours	WinCLIP	CLIP + ours
Candle	96.0	96.9±0.3	95.4	97.0±0.1	93.4±1.4	97.7±0.2	94.8±1.0	97.9±0.1	95.1±0.3	97.9±0.2
Capsules	74.9	78.1±2.3	85.0	75.6±0.3	85.0±3.1	82.1±0.6	84.9±0.8	82.5±0.6	86.8±1.7	82.6±0.6
Cashew	87.3	91.6±2.4	92.1	91.8±1.0	94.0±0.4	92.9±1.1	94.3±0.5	93.1±0.8	95.2±0.8	93.1±0.8
Chewing gum	89.8	94.3±1.3	96.5	93.0±1.9	97.6±0.8	95.0±0.7	97.3±0.8	95.1±0.7	97.7±0.3	95.3±0.6
Fryum	88.2	90.0±0.5	80.3	89.2±0.5	88.5±1.9	92.6±1.2	90.5±0.4	92.8±1.3	90.8±0.5	92.9±1.2
Macaroni1	82.0	82.5±2.0	76.2	89.4±1.4	82.9±1.5	91.5±1.7	83.3±1.9	91.5±1.7	85.2±0.9	91.6±1.6
Macaroni2	65.6	71.3±1.5	63.7	68.6±1.5	70.2±0.9	69.8±1.5	71.8±2.0	69.9±1.4	70.9±2.2	69.9±1.3
PCB1	58.2	49.7±7.0	73.6	69.4±3.8	75.6±23.0	70.7±16.9	76.7±5.2	80.2±10.4	88.3±1.7	83.1±2.5
PCB2	51.6	50.7±1.0	51.2	47.3±0.1	62.2±3.9	60.7±2.2	62.6±3.7	63.8±1.9	67.5±2.6	66.4±1.5
PCB3	66.0	67.9±1.3	73.4	66.3±0.4	74.1±1.1	74.0±8.4	78.8±1.9	80.3±3.4	83.3±1.7	83.4±1.8
PCB4	74.5	75.2±0.7	79.6	82.9±0.4	85.2±8.9	88.1±9.0	82.3±9.9	87.6±10.4	87.6±8.0	89.7±10.1
Pipe fryum	84.1	89.9±1.4	69.7	87.2±2.6	97.2±1.1	92.6±1.7	98.0±0.6	93.0±1.8	98.5±0.4	93.2±1.8

Table 9: Class-wise AUPR on the VisA dataset.

	0-shot (Object unknown)		0-shot (Object known)		1-shot		2-shot		4-shot	
Category	CLIP	CLIP + ours	WinCLIP	CLIP + ours	WinCLIP	CLIP + ours	WinCLIP	CLIP + ours	WinCLIP	CLIP + ours
Candle	96.3	96.9±0.3	95.8	96.9±0.2	93.6±1.5	97.6±0.2	95.1±1.1	97.7±0.1	95.3±0.4	97.8±0.1
Capsules	83.1	85.4±1.5	90.9	84.8±0.1	89.9±2.5	88.6±0.4	88.9±0.7	88.9±0.3	91.5±1.4	89.0±0.3
Cashew	94.3	96.2±1.0	96.4	96.3±0.4	97.2±0.2	96.9±0.4	97.3±0.2	96.9±0.3	97.7±0.4	96.9±0.3
Chewing gum	95.6	97.7±0.6	98.6	97.0±0.9	99.0±0.3	97.9±0.3	98.9±0.3	98.0±0.3	99.0±0.1	98.0±0.2
Fryum	94.8	95.7±0.2	90.1	95.4±0.3	94.7±1.0	97.0±0.5	95.8±0.2	97.0±0.6	96.0±0.3	97.1±0.6
Macaroni1	85.0	84.5±1.7	75.8	90.1±1.1	84.9±1.2	92.0±1.4	84.7±1.5	92.0±1.4	86.5±0.6	92.1±1.3
Macaroni2	60.8	67.4±1.6	60.3	66.3±1.9	68.4±1.8	68.5±1.6	70.4±1.8	68.9±1.7	69.6±2.8	68.9±1.7
PCB1	63.6	55.3±6.3	78.4	71.9±4.0	76.5±19.0	72.4±13.8	78.3±4.3	80.0±8.8	87.7±1.7	82.4±2.6
PCB2	54.9	54.0±0.7	49.2	49.6±0.1	64.9±3.3	64.7±2.1	65.8±4.0	68.0±1.5	71.3±3.4	70.1±0.7
PCB3	66.4	69.6±1.3	76.5	67.3±0.3	73.5±1.6	74.6±8.7	80.9±1.6	82.5±3.0	84.8±1.8	85.4±1.4
PCB4	79.6	79.9±0.5	77.7	84.4±0.4	78.5±15.5	83.2±13.7	72.5±16.2	83.6±14.4	85.6±8.9	86.0±14.6
Pipe fryum	91.6	95.1±0.8	82.3	93.6±2.1	98.6±0.5	96.2±1.1	99.0±0.3	96.4±1.1	99.2±0.2	96.5±1.1

Table 10: Class-wise F_1-max on the VisA dataset.

	0-shot (Object unknown)		0-shot (Object known)		1-shot		2-shot		4-shot	
Category	CLIP	CLIP + ours	WinCLIP	CLIP + ours	WinCLIP	CLIP + ours	WinCLIP	CLIP + ours	WinCLIP	CLIP + ours
Candle	91.1	92.4±0.6	89.4	92.2±0.3	87.8±1.2	94.7±0.5	89.1±1.3	95.2±0.4	88.9±1.0	95.1±0.3
Capsules	79.8	81.4±1.3	83.9	79.3±0.2	84.9±2.0	82.8±0.6	85.4±0.6	83.0±0.6	86.0±0.9	83.1±0.5
Cashew	84.3	89.0±2.3	88.4	89.1±1.2	90.7±0.7	89.6±1.3	90.9±0.7	89.8±1.0	91.6±1.3	90.0±1.0
Chewing gum	87.4	92.9±1.7	94.8	90.2±2.4	95.6±0.9	92.7±1.0	95.4±0.6	93.0±0.9	95.7±0.5	93.4±0.6
Fryum	86.2	87.6±0.8	82.7	86.9±0.9	87.2±1.4	91.0±2.0	88.4±0.6	91.0±1.9	88.9±0.8	91.4±1.9
Macaroni1	74.4	76.7±1.1	74.2	83.1±1.2	76.2±1.4	84.9±1.5	76.7±2.0	84.8±1.5	78.2±1.2	84.9±1.6
Macaroni2	71.9	71.6±1.1	69.8	69.4±0.7	72.3±1.1	70.1±0.9	73.9±0.9	69.9±0.7	73.1±1.6	70.0±0.7
PCB1	66.7	66.7±0.0	71.0	69.2±1.1	81.3±6.6	75.6±6.9	73.2±3.7	78.7±4.8	83.1±2.2	78.5±2.3
PCB2	67.1	67.2±0.2	67.1	67.1±0.1	67.2±0.3	68.1±0.8	67.3±0.3	68.5±0.5	67.7±0.6	69.3±0.8
PCB3	70.5	69.9±0.9	71.0	68.9±0.3	73.5±1.5	72.7±3.4	73.9±1.3	75.3±2.0	77.0±1.4	76.9±1.8
PCB4	73.3	74.3±0.6	74.9	77.0±0.5	86.1±2.1	89.6±3.9	86.8±3.8	88.4±6.6	84.6±7.0	89.6±5.8
Pipe fryum	85.1	88.6±1.6	80.7	86.9±1.3	94.4±0.7	90.9±1.8	95.4±0.8	91.3±1.6	95.6±0.7	91.5±1.4