

WinCLIP: Zero-/Few-Shot Anomaly Classification and Segmentation

Jongheon Jeong^{2*†} Yang Zou^{1*} Taewan Kim¹
Dongqing Zhang¹ Avinash Ravichandran^{1‡} Onkar Dabeer¹
¹ AWS AI Labs ² KAIST

Abstract

Visual anomaly classification and segmentation are vital for automating industrial quality inspection. The focus of prior research in the field has been on training custom models for each quality inspection task, which requires task-specific images and annotation. In this paper we move away from this regime, addressing zero-shot and few-normal-shot anomaly classification and segmentation. Recently CLIP, a vision-language model, has shown revolutionary generality with competitive zero-/few-shot performance in comparison to full-supervision. But CLIP falls short on anomaly classification and segmentation tasks. Hence, we propose window-based CLIP (WinCLIP) with (1) a compositional ensemble on state words and prompt templates and (2) efficient extraction and aggregation of window/patch/image-level features aligned with text. We also propose its few-normal-shot extension WinCLIP+, which uses complementary information from normal images. In MVTec-AD (and VisA), without further tuning, WinCLIP achieves 91.8%/85.1% (78.1%/79.6%) AUROC in zero-shot anomaly classification and segmentation while WinCLIP+ does 93.1%/95.2% (83.8%/96.4%) in 1-normal-shot, surpassing state-of-the-art by large margins.

1. Introduction

Visual anomaly classification (AC) and segmentation (AS) classify and localize defects in industrial manufacturing, respectively, predicting an image or a pixel as normal or anomalous. Visual inspection is a long-tail problem. The objects and their defects vary widely in color, texture, and size across a wide range of industrial domains, including aerospace, automobile, pharmaceutical, and electronics. These result in two main challenges in the field.

First, defects are rare with wide range of variations, leading to a lack of representative anomaly samples in the

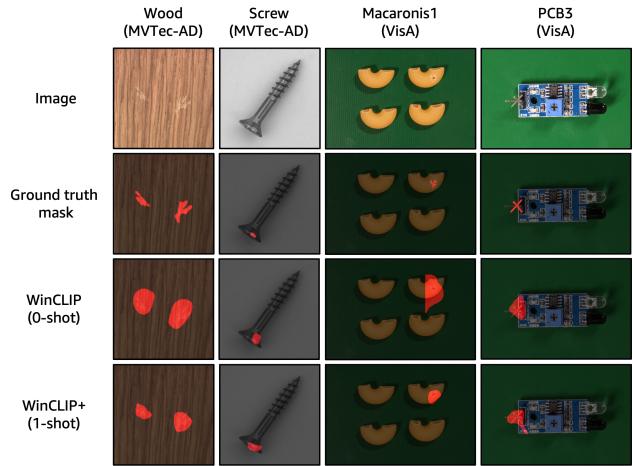


Figure 1. Language guided zero-/one-shot¹ anomaly segmentation from WinCLIP/WinCLIP+. Best viewed in color and zoom in.

training data. Consequently, existing works have mainly focused on one-class or unsupervised anomaly detection [2, 7, 8, 20, 29, 31, 53, 59], which only requires normal images. These methods typically fit a model to the normal images and treat any deviations from it as anomalous. When hundreds or thousands of normal images are available, many methods achieve high-accuracy on public benchmarks [3, 8, 31]. But in the few-normal-shot regime, there is still room to improve performance [14, 32, 39, 59], particularly in comparison with the fully-supervised upper bound.

Second, prior work has focused on training a bespoke model for each visual inspection task, which is not scalable across the long-tail of tasks. This motivates our interest in zero-shot anomaly classification and segmentation. But many defects are defined with respect to a normal image. For example, a missing component on a circuit board is most easily defined with respect to a normal circuit board with all components present. For such cases, at least a few normal images are needed. So in addition to the zero-shot case, we also consider the case of few-normal-shot anomaly classification and segmentation. Since only few normal images are available, there is no segmentation supervision for localizing anomalies, making this a challenging problem across the long-tail of tasks.

[†]Work done during an Amazon internship.

^{*}The authors contributed equally.

[‡]Work done as part of AWS AI Labs.

¹few-shot and few-normal-shot are used interchangeably in our case.

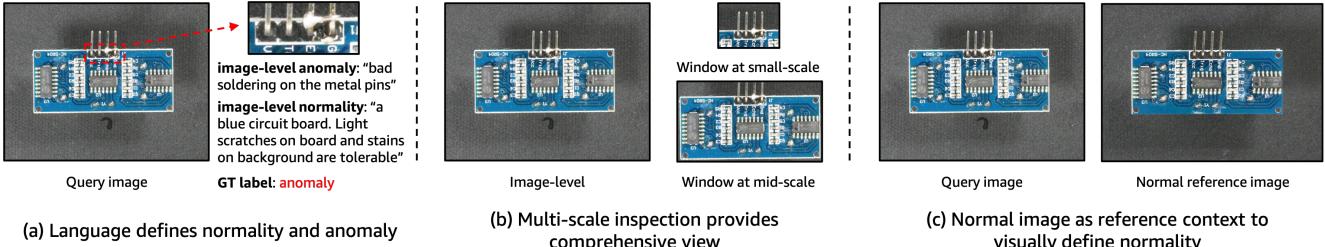


Figure 2. Motivation of language guided visual inspection. (a) Language helps describe and clarify normality and anomaly; (b) Aggregating multi-scale features helps identify local defects; (c) Normal images provide rich referencing content to visually define normality

Vision-language models [1, 18, 27, 36] have shown promise in zero-shot classification tasks. Large-scale training with vision-language annotated pairs learns expressive representations that capture broad concepts. Without additional fine-tuning, text prompts can then be used to extract knowledge from such models for zero-/few-shot transfer to downstream tasks including image classification [27], object detection [11] and segmentation [46]. Since CLIP is one of the few open-source vision-language models, these works build on top of CLIP, benefiting from its generalization ability, and showing competitive low-shot performances in both seen and unseen objects compared to full supervision.

In this paper, we focus on zero-shot and few-normal-shot (1 to 4) regime, which has received limited attention [14, 32, 39]. Our hypothesis is that language is perhaps even more important for zero-shot/few-normal-shot anomaly classification and segmentation. This hypothesis stems from multiple observations. First, “normal” and “anomalous” are states [17] of an object that are context-dependent, and language helps clarify these states. For example, “a hole in a cloth” may be a desirable or undesirable depending upon whether distressed fashion or regular fashion clothes are being manufactured. Language can bring such context and specificity to the broad “normal” and “anomalous” states. Second, language can provide additional information to distinguish defects from acceptable deviations from normality. For example, in Figure 2(a), language provides information on the soldering defect, while minor scratches/stains on background are acceptable. In spite of these advantages, we are not aware of prior work leveraging vision-language models for anomaly classification and segmentation. In this work, with the pre-trained CLIP as a base model, we show and verify our hypothesis that language aids zero-/few-shot anomaly classification/segmentation.

Since CLIP is one of the few open-source vision-language models, we build on top of it. Previously, CLIP-based methods have been applied for zero-shot classification [27]. CLIP can be applied in the same way to anomaly classification, using text prompts for “normal” and “anomalous” as classes. However, we find naïve prompts are not effective (see Table 3). So we improve the naïve baseline with a state-level

word ensemble to better describe normal and anomalous states. Another challenge is that CLIP is trained to enforce cross-modal alignment only on the global embeddings of image and text. However, for anomaly segmentation we seek pixel-level classification and it is non-trivial to extract dense visual features aligned with language for zero-shot anomaly segmentation. Therefore, we propose a new *Window-based CLIP* (WinCLIP), which extracts and aggregates the multi-scale features while ensuring vision-language alignment. The multiple scales used are illustrated in Figure 2(b). To leverage normal images available in the few-normal-shot setting, we introduce WinCLIP+, which aggregates complementary information from the language driven WinCLIP and visual cues from the normal reference images, such as the one shown in Figure 2(c). We emphasize that our zero-shot models do not require any tuning for individual cases, and the few-normal-only setup does not use any segmentation annotation, facilitating applicability across a broad range of visual inspection tasks. As a sample, Figure 1 illustrates WinCLIP and WinCLIP+ qualitative results for a few cases.

To summarize, our main contributions are:

- We introduce a compositional prompt ensemble, which improves zero-shot anomaly classification over the naïve CLIP based zero-shot classification.
- Using the pre-trained CLIP model, we propose WinCLIP, that efficiently extract and aggregate multi-scale spatial features aligned with language for zero-shot anomaly segmentation. As far as we know, we are the first to explore language-guided zero-shot anomaly classification and segmentation.
- We propose a simple reference association method, which is applied to multi-scale feature maps for image based few-shot anomaly segmentation. WinCLIP+ combines the language-guided and vision-only methods for few-normal-shot anomaly recognition.
- We show via extensive experiments on MVTec-AD and VisA benchmarks that our proposed methods WinCLIP/WinCLIP+ outperform the state-of-the-art methods in zero-/few-shot anomaly classification and segmentation with large margins.

2. Related work

Vision-language modeling. Among recent successes of large pre-trained vision-language models (VLM) [1, 18, 27], CLIP [27] is the first to perform pre-training on web-scale image-text data, showing unprecedented generality: *e.g.*, its language-driven zero-shot inference, improved both effective robustness [40] and perceptual alignment [10]. Many following VLM works explored large-scale pre-training in different aspects, *e.g.*, scaling up data [18], efficient designs [1, 21, 47], multi-tasks [22, 43], *etc.* To democratize large-scale VLM for the usages in different domains, a billion-scale data LAION-5B [36], a code base of OpenCLIP with pre-trained models [16] are open-sourced. Other works presented CLIP’s promise in zero-/few-shot transfer to downstream tasks beyond classification [11, 30, 41, 46]. Good prompt engineering and tuning can non-trivially benefit generalization performances [27, 58]. Moreover, some other works [28, 56, 57] leverage the pre-trained CLIP for language guided detection and segmentation with promising performances.

Anomaly classification and segmentation. Due to the scarcity of anomalies, the major focus has been on one-class methods with many normal images [7, 8, 20, 49, 51, 53]. While the MVTec-AD benchmark [3] is saturated by several works [31, 48, 51], their specific application is hindered due to their unscalable full-normal-shot setup. Recent works [32, 39] explored few-shot setups by leveraging augmentation to expand the small support set for better normality modeling. RegAD [14] further proposed a model-reusing by pre-training an object-agnostic registration network with diverse images to model normality for unseen object, given a few normal samples. Meanwhile, to close the gap between academical and industrial data, Visual Anomaly (VisA) [59] is introduced for a challenging benchmark over MVTec-AD. Additionally, Vision Transformer (ViT) have recently shown its potential in visual inspection [9, 25].

State classification. In some sense, anomaly classification is related to state classification [17] that predicts if an object is normal or anomalous. While the major works in computer vision focus on object, scene, or material recognition [13, 34, 38, 45], state classification aims to differentiate the fine-grained sub-object physical properties or attributes. Several datasets covering generic states/attributes (*e.g.* tall, crack, red, smooth) over diverse objects and scenes are introduced [15, 17, 23, 50]. Some works [24, 26, 44] built graphs consisting of attributes and objects, of which relationship is learnt by graph neural networks [54].

3. Background

Anomaly classification and segmentation. Given an image $\mathbf{x} \in \mathcal{X}$, both anomaly classification and segmentation (ACS) aim to predict “abnormality” in \mathbf{x} . Specifically, we consider anomaly classification (AC) as a binary classifi-

cation $\mathcal{X} \rightarrow \{-, +\}$ where “+” indicates the presence of anomaly in image-level. And anomaly segmentation (AS) is its pixel-level extension to output the location of anomalies via $\mathcal{X} \rightarrow \{-, +\}^{h \times w}$ for a certain image with size $h \times w$. In practice, the tasks are often cast into problems of predicting anomaly scores. For example, anomaly classification typically models a mapping ascore : $\mathcal{X} \rightarrow [0, 1]$ so that a binary classification can be performed by thresholding ascore(\mathbf{x}).

Due to the lack of anomalous (or positive) samples in practice, the one-class scenario, where the training data $\mathcal{D} := \{(x_i, -)\}_{i=1}^K$ consists of only normal (or negative) samples, has been widely used. In this paper, we follow the one-class protocol, particularly focusing on extreme cases of few-shot ($K = 1$ to 4) and the unexplored zero-shot setups for both AC and AS. And we assume an available list of task-specific texts tags, *e.g.*, for objects and relevant defects.

Zero-shot classification with CLIP. *Contrastive Language Image Pre-training* (CLIP) [27] is a large-scale pre-training method offering a joint vision-language representation. Given million-scale image-text pairs $\{(x_t, s_t)\}_{t=1}^T$ from the web, CLIP trains an image encoder f and a text encoder g via contrastive learning [6, 55] to maximize the correlation between $f(x_t)$ and $g(s_t)$ across t in terms of cosine similarity $\langle f(\mathbf{x}), g(\mathbf{s}) \rangle$. Given an input \mathbf{x} and a closed set of free-form texts $S = \{s_1, \dots, s_k\}$, CLIP can perform zero-shot classification via a k -way categorical distribution:

$$p(s = s_i | \mathbf{x}; \mathbf{s} \in S) := \frac{\exp(\langle f(\mathbf{x}), g(s_i) \rangle / \tau)}{\sum_{s \in S} \exp(\langle f(\mathbf{x}), g(s) \rangle / \tau)}, \quad (1)$$

where $\tau > 0$ is the temperature hyperparameter.

For a set of class words $C = \{c_1, \dots, c_k\}$, it has shown that accompanying each label word $c \in C$ with a *prompt template*, *e.g.*, “a photo of a [c]”, improves accuracy over the case without templates. Moreover, an ensemble of prompt embeddings that aggregates multiple (80) templates *e.g.*, “a cropped photo of a [c]”, can further boost the performance [27]. Overall, we are essentially “retrieving” the visual knowledge of CLIP through the language interface in appropriate manners. In this paper, we further explore how to extract the knowledge of CLIP in a way more suitable for anomaly recognition.

4. WinCLIP and WinCLIP+

In this section, we first establish a novel binary zero-shot anomaly classification framework with a Compositional Prompt Ensemble to improve CLIP for anomaly classification (Section 4.1). Next, we propose a simple-yet-effective *Window-based CLIP* (WinCLIP) for efficient zero-shot anomaly segmentation (Section 4.2). Lastly, we propose an extension *WinCLIP+* to benefit from few normal reference images, while maintaining the complementary benefits of language-guided predictions (Section 4.3).

4.1. Language-driven zero-shot AC

Two-class design. We introduce a binary zero-shot anomaly classification framework *CLIP-AC* by adapting CLIP with two class prompts $[c]$ - “normal $[o]$ ” vs. “anomalous $[o]$ ”. $[o]$ is an object-level label, *e.g.*, “bottle” when available, or simply “object”. In addition, we also test a one-class design by only using the normal prompt $s_- := \text{“normal } [o]\text{”}$ to define anomaly score as $-\langle f(\mathbf{x}), g(s_-) \rangle$. We observe the simple two-class design from CLIP already yields a non-trial performance and outperforms one-class design significantly in experiments (Table 3). This demonstrates (a) CLIP pre-trained by large web dataset provides a powerful representation with good alignment between text and images for anomaly tasks (b) specific definition about anomaly is necessary for good performance.

Compositional prompt ensemble (CPE). Unlike object-level classifiers, CLIP-AC performs classification between two *states* of a given object, *i.e.*, either “normal” or “anomalous”, which are subjective with various definitions depending on tasks. For example, “missing transistor” is “anomalous” for a circuit board while “cracked” is “anomalous” for wood. To better define the two abstract states of objects, we propose a Compositional Prompt Ensemble to generate all combinations of pre-defined lists of (a) *state words* per label and (b) *text templates*, rather than freely writing definitions. The state words include common states shared by most objects, *e.g.*, “flawless” for normality/“damaged” for anomaly. Also we can optionally add task-specific state words given prior knowledge of defects, *e.g.*, “bad soldering” on PCB. Moreover, we curate a template list specifically for anomaly tasks *e.g.*, “a photo of a $[c]$ for visual inspection”. Check details on prompt engineering in supplementary. As in top-left of Figure 4, after getting all the combinations of states and templates, we compute the average of text embeddings per label to represent the normal and anomalous classes. Note that CPE is different from CLIP prompt ensemble that does not explain object labels (*e.g.*, “cat”) and only augments templates selected by trial-and-error for object classification, including the ones unsuitable for anomaly tasks, *e.g.*, “a cartoon $[c]$ ”. Thus, the texts from CPE are more aligned with images in CLIP’s joint embedding space for anomaly tasks. We denote the zero-shot scoring model with CPE as $\text{ascore}_0 : \mathbb{R}^d \rightarrow [0, 1]$ for an image embedding $f(\mathbf{x})$.

Remark. Our two-class design with CPE is a novel approach to define anomaly compared to standard one-class methods [31, 33]. Anomaly detection is an ill-posed problem due to the open-ended nature. Previous methods model normality only by normal images regarding any deviation from normality, *e.g.*, “scratch on circuit” vs. “tiny yet acceptable scratch”. But language can define states in concrete words.

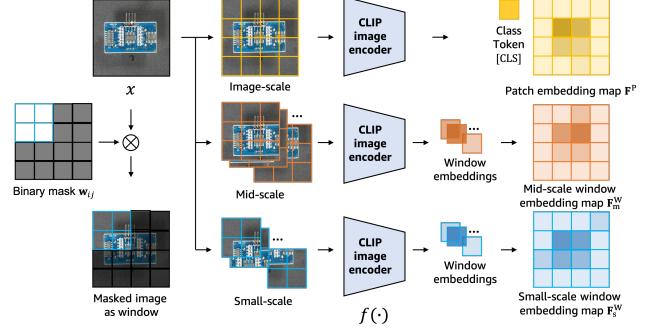


Figure 3. WinCLIP feature extraction in multiple scales of windows through CLIP image encoder, *e.g.*, ViT taking a sequence of (non-masked) patches as input. Window embeddings encode the global information (*e.g.*, from the class token) within each window.

4.2. WinCLIP for zero-shot AS

Given the language guided anomaly scoring model from CPE, we propose Window-based CLIP (WinCLIP) for zero-shot anomaly segmentation to predict pixel-level anomalies. WinCLIP extracts dense visual features with good language alignment and local details for \mathbf{x} , followed by applying ascore_0 spatially to obtain the anomaly segmentation map. Specifically, given an image \mathbf{x} of resolution $h \times w$ and an image encoder f , WinCLIP obtains a map of d -dimensional feature map $\mathbf{F}^W \in \mathbb{R}^{h \times w \times d}$ as follows:

1. Generate a set of sliding windows $\{\mathbf{w}_{ij}\}_{ij}$, where each window $\mathbf{w}_{ij} \in \{0, 1\}^{h \times w}$ is a binary mask that is active locally for a $k \times k$ kernel around (i, j) .
2. Collect each output embedding \mathbf{F}_{ij}^W , computed from the active area of \mathbf{x} after applying each \mathbf{w}_{ij} , defined by:

$$\mathbf{F}_{ij}^W := f(\mathbf{x} \odot \mathbf{w}_{ij}), \quad (2)$$

where \odot is the element-wise product (see Figure 3).

Figure 3 illustrates the dense feature extraction of WinCLIP with ViT while it is also applicable to CNN.

In addition, we also explore a natural dense representation candidate, *penultimate feature map*, the last feature map before pooling. Specifically, for patch embedding map \mathbf{F}^P (other than the *class token* $[\text{CLS}]$) of ViT-based CLIP, top of Figure 3, we apply ascore_0 patch-wisely for segmentation. However, we observe that such patch-level features are not aligned with the language space, leading to a poor dense predictions (Table 8). We conjecture this is caused by those features have not been directly supervised with language signal in CLIP. Also these patch features have already aggregated the global context due to self-attention, hindering capturing local details for segmentation.

Compared to the penultimate features \mathbf{F}^P , we remark dense features from WinCLIP is more aligned with language: *e.g.*, for ViT-based CLIP, all the features in \mathbf{F}^W are now from

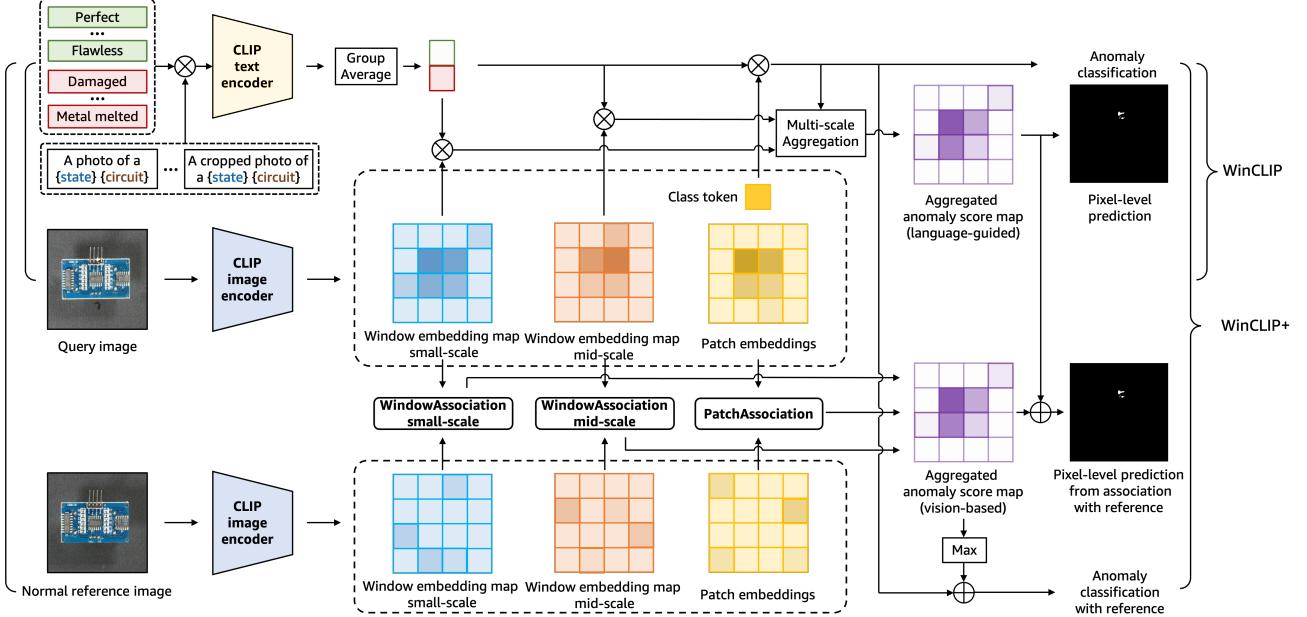


Figure 4. Workflows of WinCLIP/WinCLIP+ (upper/entire pane). Various states and templates are composed and converted to two text embeddings as class prototypes via CLIP text encoder (Section 4.1). The class prototypes are correlated with the multi-scale features from CLIP image encoder (Figure 3) for zero-shot AC/AS in WinCLIP. WinCLIP+ applies the reference association on patch, small-/mid-window (Patch/WindowAssociation) for vision-based anomaly score maps, which are aggregated for few-shot AS/AC with language-guided scores.

class tokens which are directly aligned to texts in CLIP pre-training. Also the features focus more on local details via sliding windows. Lastly, WinCLIP can be efficiently computed, especially with ViT architecture. Concretely, the computation of (2) can directly benefit from just dropping all the masked patches before forwarding them, in a similar manner to masked autoencoder [12].

Harmonic aggregation of windows. For each local window, the zero-shot anomaly score $M_{0,ij}^w$ is similarity between the window feature F_{ij}^w and text embeddings from compositional prompt ensemble. This score is distributed to every pixel of the local window. Then at each pixel, we aggregate multiple scores from all overlapping windows to improve segmentation by *harmonic averaging* (3), weighting more on scores towards normality prediction (zero value).

$$\bar{M}_{0,ij}^w := \left(\frac{1}{\sum_{u,v} (\mathbf{w}_{uv})_{ij}} \sum_{u,v} \frac{(\mathbf{w}_{uv})_{ij}}{M_{0,uv}^w} \right)^{-1}. \quad (3)$$

Multi-scale aggregation. The kernel size k corresponds to the amount of surrounding context for each location in computing WinCLIP features (2). It controls the balance between local details and global information in segmentation. To capture defects of sizes ranging from small to large scale, we aggregate predictions from multi-scale features: *e.g.*, (a) small-scale (2×2 in patch scales of ViT; corresponds to 32×32 in pixels), (b) mid-scale (3×3 in ViT; 48×48), and (c) image-scale feature (ViT class token capturing image context

due to self-attention). We also adopt harmonic averaging for aggregation. Figure 3 illustrates the features on each scale.

4.3. WinCLIP+ with few-normal-shots

For a comprehensive anomaly classification and segmentation, language guided zero-shot approach is not enough as certain defects can only be defined via visual reference rather than only text. For example, “Metal-nut” in MVTec-AD [3] has an anomaly type labeled as “flipped upside-down”, which can only be identified relatively from a normal image. To define and recognize the anomalies more precisely, we propose an extension of WinCLIP, *WinCLIP+*, by incorporating K normal reference images $\mathcal{D} := \{(x_i, -)\}_{i=1}^K$. WinCLIP+ combines the complementary prediction from both language-guided and visual based approaches for better anomaly classification and segmentation.

We first propose a *reference association* as the key module to incorporate given reference images, which can simply store and retrieve the memory features \mathbf{R} of \mathcal{D} based on the cosine similarity. Given such module and the corresponding (*e.g.*, patch-level²) features $\mathbf{F} \in \mathbb{R}^{h \times w \times d}$ extracted from a query image, a prediction $\mathbf{M} \in [0, 1]^{h \times w}$ for anomaly segmentation can be made by:

$$\mathbf{M}_{ij} := \min_{r \in \mathbf{R}} \frac{1}{2}(1 - \langle \mathbf{F}_{ij}, r \rangle). \quad (4)$$

Then we apply this association module at multiple scales of feature maps that are obtained from WinCLIP (see Fig-

²Nevertheless, the module is generally applicable for other scales.

ure 4 for the overall illustration). Specifically, given few-shot samples, we construct separate reference memories from three different features: (a) WinCLIP features at small-scale \mathbf{F}_s^W , (b) those at mid-scale \mathbf{F}_m^W , and also (c) from penultimate features \mathbf{F}^P with global context (*e.g.*, the patch tokens in ViT capturing image context due to self-attention). Even though \mathbf{F}^P is not aligned with language, it still useful to define normality and anomaly.

As a result, WinCLIP+ gets three reference memories: \mathbf{R}_s^W , \mathbf{R}_m^W , and \mathbf{R}^P . Then, we average their multi-scale predictions (4) for anomaly segmentation for a given query,

$$\mathbf{M}^W := \frac{1}{3}(\mathbf{M}^P + \mathbf{M}_s^W + \mathbf{M}_m^W), \quad (5)$$

and then fusing with our language-guided prediction $\bar{\mathbf{M}}_0^W$.

To perform anomaly classification, we combine the maximum value of \mathbf{M}^W and the WinCLIP zero-shot classification score. The two scores have complementary information to collaborative with, specifically (a) one from the spatial features of few-shot references, and (b) the other one from the CLIP knowledge retrieved via language:

$$\text{ascore}_W(\mathbf{x}) := \frac{1}{2} \left(\text{ascore}_0(f(\mathbf{x})) + \max_{ij} \mathbf{M}_{ij}^W \right). \quad (6)$$

5. Experiments

We perform an array of experiments to evaluate the performance of WinCLIP-based ACS under low-shot regimes, covering recent challenging benchmarks on industrial anomaly classification and segmentation that we are focusing on. We also conduct an extensive ablation study to validate the individual effectiveness of our proposed components. The detailed setups, *e.g.*, pre-processing, metrics, and other implementation details, are given in the supplementary.

Datasets. Our experiments are based on MVTec-AD [3] and VisA [59] datasets. Both benchmarks have diverse subsets of different objects, *e.g.*, capsules, circuit boards. They contain high-resolution images (*e.g.*, 700^2 - 1024^2 for MVTec-AD, and roughly $1.5K \times 1K$ for VisA) of common objects with the full pixel-level annotations.

Evaluation metrics. For classification, we report (a) *Area Under the Receiver Operating Characteristic* (AUROC) following the literature [8, 31, 49], as well as (b) *Area Under the Precision-Recall curve* (AUPR) and (c) *F₁-score at optimal threshold* ($F_1\text{-max}$) for a clearer view against potential data imbalance [59]). For segmentation, we report (a) *pixel-wise AUROC* (pAUROC) and (b) *Per-Region Overlap* (PRO) [4] scores [8, 20], and (c) (*pixel-wise*) $F_1\text{-max}$ in a similar manner to the anomaly classification evaluation.

Implementation details. We adopt the CLIP implementation of OpenCLIP³ and its public pre-trained models in our

experiments: namely, we use the LAION-400M [37] based CLIP with ViT-B/16+ [16] unless otherwise noted. We apply WinCLIP with stride 1 on ViT patch embeddings, which is equivalent to stride 16 in pixel-level in case of ViT-B/16+.

5.1. Zero-/few-shot anomaly classification

In Table 1 we compare zero-shot and few-normal-shot anomaly classification results with prior works.

For zero-shot setup, we compare WinCLIP with two prior models: CLIP-AC (first row of Table 1), which is the original CLIP zero-shot classification [27] with labels of the form {“normal [c]”, “anomalous [c]”}, and CLIP-AC with the prompt ensemble (second row in Table 1) from [27] engineered for ImageNet [19]. We see that WinCLIP significantly improves over using these naïve adaptations of CLIP on both MVTec-AD and VisA. Section 5.4 presents ablation study on a break-down of this gain.

For the few-normal-shot setup, we see the same trend: WinCLIP+ outperforms prior works by a wide margin across all metrics on both benchmarks. In particular, we improve upon the state-of-the-art PatchCore [31] by 9.7% on 1-shot MVTec-AD and by 5.3% on 1-shot VisA. On MVTec-AD, we note that zero-shot WinCLIP outperforms the few-shot versions of prior works. Furthermore, WinCLIP+ 1/2/4-shot performance is better than WinCLIP 0-shot performance, highlighting the additional value of reference normal images.

5.2. Zero-/few-shot anomaly segmentation

In Table 4 we compare zero-shot and few-normal-shot anomaly segmentation results with prior works. While there are no prior works on zero-shot anomaly segmentation, we adapt two methods developed for other problems to our setup. First, Trans-MM [5] is a recent model interpretation method applicable to Transformers that provides a pixel-level mask. Second, MaskCLIP [57] is a general semantic segmentation model based on CLIP. We see that WinCLIP outperforms both methods by a wide margin on both MVTec-AD and VisA, highlighting that generic adaptations of CLIP do not perform as well as WinCLIP.

For the few-normal-shot setup, we compare with three prior works, which are designed specifically for anomaly localization. We see that WinCLIP+ again outperforms these prior methods across all metrics on both benchmarks, showing the additional value provided by language prompts. In Figure 5, we show qualitative results for a number of objects and defects. We see that in all cases, 1-shot WinCLIP+ provides a mask that is more concentrated on the ground truth compared to prior works. We also see that 1/2/4-normal-shot WinCLIP+ is better than 0-shot WinCLIP, demonstrating the complementary benefits of language driven prediction and visual only based model based on reference normal images.

³https://github.com/mlfoundations/open_clip

Anomaly Classification		MVTec-AD			VisA		
Setup	Method	AUROC	AUPR	F_1 -max	AUROC	AUPR	F_1 -max
0-shot	CLIP-AC [27]	74.0 \pm 0.0	89.1 \pm 0.0	88.5 \pm 0.0	59.3 \pm 0.0	67.0 \pm 0.0	74.4 \pm 0.0
	+ Prompt ens. [27]	74.1 \pm 0.0	89.5 \pm 0.0	87.8 \pm 0.0	58.2 \pm 0.0	66.4 \pm 0.0	74.0 \pm 0.0
	WinCLIP (ours)	91.8\pm0.0	96.5\pm0.0	92.9\pm0.0	78.1\pm0.0	81.2\pm0.0	79.0\pm0.0
1-shot	SPADE [7]	81.0 \pm 2.0	90.6 \pm 0.8	90.3 \pm 0.8	79.5 \pm 4.0	82.0 \pm 3.3	80.7 \pm 1.9
	PaDiM [8]	76.6 \pm 3.1	88.1 \pm 1.7	88.2 \pm 1.1	62.8 \pm 5.4	68.3 \pm 4.0	75.3 \pm 1.2
	PatchCore [31]	83.4 \pm 3.0	92.2 \pm 1.5	90.5 \pm 1.5	79.9 \pm 2.9	82.8 \pm 2.3	81.7 \pm 1.6
2-shot	WinCLIP+ (ours)	93.1\pm2.0	96.5\pm0.9	93.7\pm1.1	83.8\pm4.0	85.1\pm4.0	83.1\pm1.7
	SPADE [7]	82.9 \pm 2.6	91.7 \pm 1.2	91.1 \pm 1.0	80.7 \pm 5.0	82.3 \pm 4.3	81.7 \pm 2.5
	PaDiM [8]	78.9 \pm 3.1	89.3 \pm 1.7	89.2 \pm 1.1	67.4 \pm 5.1	71.6 \pm 3.8	75.7 \pm 1.8
4-shot	PatchCore [31]	86.3 \pm 3.3	93.8 \pm 1.7	92.0 \pm 1.5	81.6 \pm 4.0	84.8 \pm 3.2	82.5 \pm 1.8
	WinCLIP+ (ours)	94.4\pm1.3	97.0\pm0.7	94.4\pm0.8	84.6\pm2.4	85.8\pm2.7	83.0\pm1.4
	SPADE [7]	84.8 \pm 2.5	92.5 \pm 1.2	91.5 \pm 0.9	81.7 \pm 3.4	83.4 \pm 2.7	82.1 \pm 2.1
	PaDiM [8]	80.4 \pm 2.5	90.5 \pm 1.6	90.2 \pm 1.2	72.8 \pm 2.9	75.6 \pm 2.2	78.0 \pm 1.2
	PatchCore [31]	88.8 \pm 2.6	94.5 \pm 1.5	92.6 \pm 1.6	85.3 \pm 2.1	87.5 \pm 2.1	84.3 \pm 1.3
	WinCLIP+ (ours)	95.2\pm1.3	97.3\pm0.6	94.7\pm0.8	87.3\pm1.8	88.8\pm1.8	84.2\pm1.6

Table 1. Comparison of anomaly classification (AC) performance on MVTec-AD and VisA benchmarks. We report the mean and standard deviation over 5 random seeds for each measurement. Bold indicates the best performance.

5.3. Comparison with many-shot methods

In Table 2 we compare our zero-/few-shot results with full-shot results of several prior works on MVTec-AD. Our 4-shot WinCLIP+ is competitive with CutPaste [20], a recent method that utilizes the *full-shot* samples for model tuning. Also, our 0-shot WinCLIP outperforms recent few-shot methods in AC, such as DifferNet [32] and TDG [39], even compared to their results with more than 10-shots. Recently, a new setup of aggregated few-shot is proposed [14], where one is free to use all the training samples but for the target class which is restricted to k -shot. Our 4-shot WinCLIP+ outperforms RegAD’s aggregated 4-shot [14] performance.

5.4. Ablation study

We perform component-wise analysis on MVTec-AD [3]. A further study, *e.g.*, comparison with CLIP-based PatchCore, effect of different backbones, discussion on failure cases, *etc.*, can be found in the supplementary material.

WinCLIP for AC: In Table 3, we report the individual effect of components that constitute our zero-shot AC model. Firstly, we observe (a) the textual supervision for the word “anomalous” is crucial to achieve a reasonable performance (“One-class”; Section 4.1), suggesting the effectiveness of CLIP knowledge about “abnormality”. Next, we confirm that having a diversity in both (b) state-level and (c) prompt-level texts are the key source of gains. And we remark the proposed state ensemble as a more significant component. Finally, we observe (d) applying multi-crop prediction [13] could also yield a minor improvement.

WinCLIP for AS: Table 8 validates not only the efficiency

Methods	Setup	AC	AS
WinCLIP (ours)	0-shot	91.8	85.1
WinCLIP+ (ours)	1-shot	93.1	95.2
WinCLIP+ (ours)	4-shot	95.2	96.2
DifferNet [32]	16-shot	87.3	-
TDG [39]	10-shot	78.0	-
RegAD-L [14]	2-shot	81.5	93.3
RegAD [14]	4 + agg.	88.2	95.8
MKD [35]	full-shot	87.7	90.7
P-SVDD [49]	full-shot	92.1	95.7
CutPaste [20]	full-shot	95.2	96.0
PatchCore [31]	full-shot	99.6	98.2

Table 2. Comparison with existing many-shot ACS methods in AUROC (or pixel-) on MVTec-AD.

Method	AUROC	AUPR	F_1 -max
(a) One-class	34.2	68.9	83.5
Two-class	74.0	89.1	88.5
(b) + State ens.	89.8	95.6	92.2
(c) + Prompt ens.	90.8	96.1	92.5
(d) + Multi-crop	91.8	96.5	92.9

Table 3. Comparison of AC performance on MVTec-AD across WinCLIP ablations in AC (Section 4.1).

of WinCLIP to extract local features for zero-shot AS, but also the effectiveness of multi-scale and harmonic averaging to boost the results. To this end, we consider the following additional baselines that also extract patch-level features: (i) *Patch-token* (Section 4.2): it takes the patch features at the last layer, and (ii) *Image tiling*: it first performs dense “tiling” on an image and then obtains “tile” embeddings for segmentation by forwarding each tile with resizing. Overall, the comparison shows that patch-tokens are not aligned with language despite its fast inference time, while “Image tiling” makes a significant computational overhead although it does benefit from their local features. WinCLIP achieves accelerated inference due to its window-based computation of local features, with even better performance. Also based on the multi-scale study, we observe that segmentation benefits from both features with image-level, and middle/local context. Note that the scores from last patch embeddings of ViT encodes global context thanks to self-attention, which contributes to a comprehensive localization in WinCLIP.

WinCLIP+ for AC and AS: We ablate on different factors to define WinCLIP+ scores for AC (6) and AS (5) respectively. For AC, from Table 5, we clearly remark the effectiveness of ascore_0 upon $\max \mathbf{M}^W$. Interestingly, we observe ascore_0 is beneficial even in higher-shot regimes where $\max \mathbf{M}^W$ can be better, confirming their complementary effects. For AS, in Table 6, we notice the effect of adding \mathbf{M}_m^W (or \mathbf{M}_s^W) upon \mathbf{M}^P , *i.e.*, the prediction from WinCLIP features: apart from the good performance of \mathbf{M}^P , \mathbf{M}^W could still provide useful information from its local-awareness.

WinCLIP with task-specific defects: As mentioned in Section 4.1, besides using the generic state words and tem-

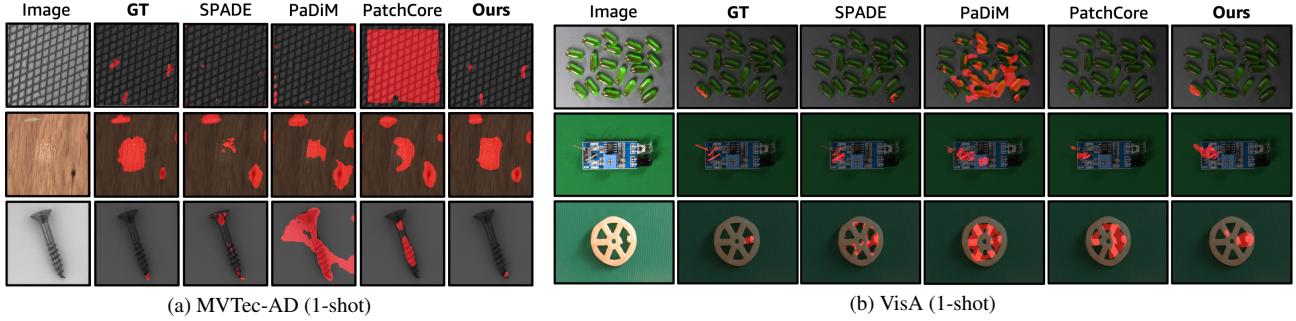


Figure 5. Qualitative comparison of 1-shot anomaly segmentation results on MVTec-AD and VisA benchmarks.

Anomaly Segmentation		MVTec-AD			VisA		
Setup	Method	pAUROC	PRO	F_1 -max	pAUROC	PRO	F_1 -max
0-shot	Trans-MM [5]	57.5±0.0	21.9±0.0	12.1±0.0	49.4±0.0	10.2±0.0	3.1±0.0
	MaskCLIP [57]	63.7±0.0	40.5±0.0	18.5±0.0	60.9±0.0	27.3±0.0	7.3±0.0
	WinCLIP (ours)	85.1±0.0	64.6±0.0	31.7±0.0	79.6±0.0	56.8±0.0	14.8±0.0
1-shot	SPADE [7]	91.2±0.4	83.9±0.7	42.4±1.0	95.6±0.4	84.1±1.6	35.5±2.2
	PaDiM [8]	89.3±0.9	73.3±2.0	40.2±2.1	89.9±0.8	64.3±2.4	17.4±1.7
	PatchCore [31]	92.0±1.0	79.7±2.0	50.4±2.1	95.4±0.6	80.5±2.5	38.0±1.9
2-shot	WinCLIP+ (ours)	95.2±0.5	87.1±1.2	55.9±2.7	96.4±0.4	85.1±2.1	41.3±2.3
	SPADE [7]	92.0±0.3	85.7±0.7	44.5±1.0	96.2±0.4	85.7±1.1	40.5±3.7
	PaDiM [8]	91.3±0.7	78.2±1.8	43.7±1.5	92.0±0.7	70.1±2.6	21.1±2.4
4-shot	PatchCore [31]	93.3±0.6	82.3±1.3	53.0±1.7	96.1±0.5	82.6±2.3	41.0±3.9
	WinCLIP+ (ours)	96.0±0.3	88.4±0.9	58.4±1.7	96.8±0.3	86.2±1.4	43.5±3.3
	SPADE [7]	92.7±0.3	87.0±0.5	46.2±1.3	96.6±0.3	87.3±0.8	43.6±3.6
8-shot	PaDiM [8]	92.6±0.7	81.3±1.9	46.1±1.8	93.2±0.5	72.6±1.9	24.6±1.8
	PatchCore [31]	94.3±0.5	84.3±1.6	55.0±1.9	96.8±0.3	84.9±1.4	43.9±3.1
	WinCLIP+ (ours)	96.2±0.3	89.0±0.8	59.5±1.8	97.2±0.2	87.6±0.9	47.0±3.0

Table 4. Comparison of anomaly segmentation (AS) performance on MVTec-AD and VisA benchmarks. We report the mean and standard deviation over 5 random seeds for each measurement. Bold indicates the best performance.

Method	pAUROC	PRO	F_1 -max	Time (ms)
Patch-token	22.4	2.3	8.0	95.5±18.8
Image tiling	77.9	57.5	25.5	1442.1±62.2
WinCLIP (ours)	85.1	64.6	31.7	389.4±18.5
w/o image-scale	82.0	63.0	29.5	378.6±20.2
w/o mid-scale	84.0	61.6	30.5	<u>190.7±13.9</u>
w/o small-scale	<u>84.7</u>	<u>63.6</u>	<u>30.6</u>	265.4±15.9
w/o Harmonic avg.	81.5	60.5	27.3	279.9±22.8

Table 8. Comparison of AS performance on MVTec-AD and its per-image inference time, measured at Amazon EC2 G4dn instances.

plates (Fig. 6 of supplementary) to cover common cases, our compositional prompt ensemble also supports task-specific state words, *e.g.*, “missing part” on PCB/“burnt” pipe fryum; both VisA and MVTec-AD release specific defect types. Ablation study in Table 7 shows that specific state words further improve zero-shot classification in VisA by 0.8% average AUROC with 5.3% gain on the challenging PCB2.

6. Conclusion

We propose a novel framework to define normality and anomaly via both fine-grained textual definitions and normal

WinCLIP+ (AC)		# shots (AUROC)			
max M^w	ascore ₀	1	2	4	8
✓	✗	87.9	91.0	92.6	94.5
✗	✓	91.8	91.8	91.8	91.8
✓	✓	93.1	94.4	95.2	96.3

Table 5. k -shot AC ablations: MVTec-AD. Bold/underline indicate the best/runner-up.

WinCLIP+ (AS)			# shots (pAUROC)			
M^p	M_m^w	M_s^w	1	2	4	8
✓	✗	✗	94.5	94.8	95.4	95.8
✓	✓	✗	95.1	<u>95.7</u>	96.3	96.6
✓	✓	✓	95.2	96.0	<u>96.2</u>	<u>96.5</u>

Table 6. k -shot AS ablations: MVTec-AD. Bold/underline indicate the best/runner-up.

Method	PCB2	PCB4	Pipe fryum	Mean
WinCLIP	51.2	79.6	69.7	78.1
+ specific states	56.5	82.7	70.4	78.9

Table 7. Ablation on specific states: VisA.

reference images for comprehensive anomaly classification and segmentation. First, we show that the CLIP pre-trained on large-scale web data provides a powerful representation with good alignment between texts and images for anomaly recognition tasks. The compositional prompt ensemble defines the normality and anomaly in text and helps to distill knowledge from the pre-trained CLIP for better zero-shot anomaly recognition. WinCLIP efficiently aggregates multi-scale features with image-text alignment from window and image-level to perform zero-shot segmentation. Moreover, given a few normal samples, vision based reference association provides complementary information about the two states to language definitions, leading to few-shot WinCLIP+. In recent benchmarks, WinCLIP and WinCLIP+ outperform state-of-the-arts in zero-/few-shot setups with considerable margins. We believe our work will bring values complementary to standard one-class methods. For further improvement, vision-language pre-training with industrial domain data is a promising direction that is left as a future work.

References

- [1] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katie Millican, Malcolm Reynolds, et al. Flamingo: A visual language model for few-shot learning. *arXiv preprint arXiv:2204.14198*, 2022. 2, 3
- [2] 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. 1, 17
- [3] 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. 1, 3, 5, 6, 7
- [4] Paul Bergmann, Michael Fauser, David Sattlegger, and Carsten Steger. Uninformed students: Student-teacher anomaly detection with discriminative latent embeddings. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4183–4192, 2020. 6
- [5] Hila Chefer, Shir Gur, and Lior Wolf. Generic attention-model explainability for interpreting bi-modal and encoder-decoder Transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 397–406, 2021. 6, 8
- [6] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International Conference on Machine Learning*, pages 1597–1607. PMLR, 2020. 3
- [7] Niv Cohen and Yedid Hoshen. Sub-image anomaly detection with deep pyramid correspondences. *arXiv preprint arXiv:2005.02357*, 2020. 1, 3, 7, 8, 13
- [8] Thomas Defard, Aleksandr Setkov, Angelique Loesch, and Romaric Audiger. PaDiM: a patch distribution modeling framework for anomaly detection and localization. In *International Conference on Pattern Recognition*, pages 475–489. Springer, 2021. 1, 3, 6, 7, 8, 13
- [9] Stanislav Fort, Jie Ren, and Balaji Lakshminarayanan. Exploring the limits of out-of-distribution detection. *Advances in Neural Information Processing Systems*, 34:7068–7081, 2021. 3
- [10] Gabriel Goh, Nick Cammarata, Chelsea Voss, Shan Carter, Michael Petrov, Ludwig Schubert, Alec Radford, and Chris Olah. Multimodal neurons in artificial neural networks. *Distill*, 2021. <https://distill.pub/2021/multimodal-neurons>. 3
- [11] Xiuye Gu, Tsung-Yi Lin, Weicheng Kuo, and Yin Cui. Open-vocabulary object detection via vision and language knowledge distillation. In *International Conference on Learning Representations*, 2021. 2, 3
- [12] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16000–16009, 2022. 5
- [13] Geoffrey E Hinton, Alex Krizhevsky, and Ilya Sutskever. ImageNet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, 25(1106–1114):1, 2012. 3, 7
- [14] Chaoqin Huang, Haoyan Guan, Aofan Jiang, Ya Zhang, Michael Spratlin, and Yanfeng Wang. Registration based few-shot anomaly detection. In *European Conference on Computer Vision*, 2022. 1, 2, 3, 7
- [15] Drew A Hudson and Christopher D Manning. GQA: A new dataset for real-world visual reasoning and compositional question answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6700–6709, 2019. 3
- [16] 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*. Zenodo, July 2021. 3, 6, 12
- [17] Phillip Isola, Joseph J Lim, and Edward H Adelson. Discovering states and transformations in image collections. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1383–1391, 2015. 2, 3
- [18] Chao Jia, Yinfai Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *International Conference on Machine Learning*, pages 4904–4916. PMLR, 2021. 2, 3
- [19] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. ImageNet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017. 6
- [20] Chun-Liang Li, Kihyuk Sohn, Jinsung Yoon, and Tomas Pfister. CutPaste: Self-supervised learning for anomaly detection and localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9664–9674, 2021. 1, 3, 6, 7
- [21] Junnan Li, Ramprasaath Selvaraju, Akhilesh Gotmare, Shafiq Joty, Caiming Xiong, and Steven Chu Hong Hoi. Align before fuse: Vision and language representation learning with momentum distillation. *Advances in neural information processing systems*, 34:9694–9705, 2021. 3
- [22] Jiasen Lu, Christopher Clark, Rowan Zellers, Roozbeh Mottaghi, and Aniruddha Kembhavi. Unified-IO: A unified model for vision, language, and multi-modal tasks. *arXiv preprint arXiv:2206.08916*, 2022. 3
- [23] M Mancini, MF Naeem, Y Xian, and Zeynep Akata. Open world compositional zero-shot learning. In *34th IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2021. 3
- [24] Massimiliano Mancini, Muhammad Ferjad Naeem, Yongqin Xian, and Zeynep Akata. Learning graph embeddings for open world compositional zero-shot learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022. 3
- [25] Pankaj Mishra, Riccardo Verk, Daniele Fornasier, Claudio Picarelli, and Gian Luca Foresti. VT-ADL: A vision transformer network for image anomaly detection and localization. In *30th IEEE/IES International Symposium on Industrial Electronics (ISIE)*, June 2021. 3

- [26] MF Naeem, Y Xian, F Tombari, and Zeynep Akata. Learning graph embeddings for compositional zero-shot learning. In *34th IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2021. 3
- [27] 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. 2, 3, 6, 7
- [28] Yongming Rao, Wenliang Zhao, Guangyi Chen, Yansong Tang, Zheng Zhu, Guan Huang, Jie Zhou, and Jiwen Lu. Denseclip: Language-guided dense prediction with context-aware prompting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 3
- [29] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022. 1
- [30] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2021. 3
- [31] Karsten Roth, Latha Pemula, Joaquin Zepeda, Bernhard Schölkopf, Thomas Brox, and Peter Gehler. Towards total recall in industrial anomaly detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14318–14328, 2022. 1, 3, 4, 6, 7, 8, 13
- [32] Marco Rudolph, Bastian Wandt, and Bodo Rosenhahn. Same same but DifferNet: Semi-supervised defect detection with normalizing flows. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1907–1916, 2021. 1, 2, 3, 7
- [33] Lukas Ruff, Robert A. Vandermeulen, Nico Görnitz, Lucas Deecke, Shoaib A. Siddiqui, Alexander Binder, Emmanuel Müller, and Marius Kloft. Deep one-class classification. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80, pages 4393–4402, 2018. 4
- [34] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. ImageNet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015. 3
- [35] 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
- [36] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade W Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, Patrick Schramowski, Srivatsa R Kundurthy, Katherine Crowson, Ludwig Schmidt, Robert Kaczmarczyk, and Jenia Jitsev. LAION-5B: An open large-scale dataset for training next generation image-text models. In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2022. 2, 3
- [37] Christoph Schuhmann, Richard Vencu, Romain Beaumont, Robert Kaczmarczyk, Clayton Mullis, Aarush Katta, Theo Coombes, Jenia Jitsev, and Aran Komatsuaki. LAION-400M: Open dataset of CLIP-filtered 400 million image-text pairs. *arXiv preprint arXiv:2111.02114*, 2021. 6
- [38] Lavanya Sharan, Ce Liu, Ruth Rosenholtz, and Edward H Adelson. Recognizing materials using perceptually inspired features. *International Journal of Computer Vision*, 103(3):348–371, 2013. 3
- [39] Shelly Sheynin, Sagie Benaim, and Lior Wolf. A hierarchical transformation-discriminating generative model for few shot anomaly detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8495–8504, 2021. 1, 2, 3, 7
- [40] Rohan Taori, Achal Dave, Vaishaal Shankar, Nicholas Carlini, Benjamin Recht, and Ludwig Schmidt. Measuring robustness to natural distribution shifts in image classification. *Advances in Neural Information Processing Systems*, 33:18583–18599, 2020. 3
- [41] Yoad Tewel, Yoav Shalev, Idan Schwartz, and Lior Wolf. ZeroCap: Zero-shot image-to-text generation for visual-semantic arithmetic. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17918–17928, 2022. 3
- [42] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning*, pages 10347–10357. PMLR, 2021. 12
- [43] Peng Wang, An Yang, Rui Men, Junyang Lin, Shuai Bai, Zhikang Li, Jianxin Ma, Chang Zhou, Jingren Zhou, and Hongxia Yang. OFA: Unifying architectures, tasks, and modalities through a simple sequence-to-sequence learning framework. In *International Conference on Machine Learning*, pages 23318–23340. PMLR, 2022. 3
- [44] Max Welling and Thomas N Kipf. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*, 2017. 3
- [45] Jianxiong Xiao, James Hays, Krista A Ehinger, Aude Oliva, and Antonio Torralba. SUN database: Large-scale scene recognition from abbey to zoo. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 3485–3492. IEEE, 2010. 3
- [46] 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 *European Conference on Computer Vision*, pages 736–753. Springer, 2022. 2, 3
- [47] Jianwei Yang, Chunyuan Li, Pengchuan Zhang, Bin Xiao, Ce Liu, Lu Yuan, and Jianfeng Gao. Unified contrastive learning in image-text-label space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19163–19173, 2022. 3
- [48] Minghui Yang, Peng Wu, Jing Liu, and Hui Feng. MemSeg: A semi-supervised method for image surface defect detection using differences and commonalities. *arXiv preprint arXiv:2205.00908*, 2022. 3

- [49] Jihun Yi and Sungroh Yoon. Patch SVDD: Patch-level SVDD for anomaly detection and segmentation. In *Proceedings of the Asian Conference on Computer Vision*, 2020. 3, 6, 7
- [50] Aron Yu and Kristen Grauman. Fine-grained visual comparisons with local learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 192–199, 2014. 3
- [51] Jiawei Yu, Ye Zheng, Xiang Wang, Wei Li, Yushuang Wu, Rui Zhao, and Liwei Wu. Fastflow: Unsupervised anomaly detection and localization via 2d normalizing flows. *arXiv preprint arXiv:2111.07677*, 2021. 3
- [52] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In Edwin R. Hancock Richard C. Wilson and William A. P. Smith, editors, *Proceedings of the British Machine Vision Conference (BMVC)*, pages 87.1–87.12. BMVA Press, September 2016. 13
- [53] Vitjan Zavrtanik, Matej Kristan, and Danijel Skočaj. DRAEM – A discriminatively trained reconstruction embedding for surface anomaly detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8330–8339, 2021. 1, 3
- [54] Si Zhang, Hanghang Tong, Jiejun Xu, and Ross Maciejewski. Graph convolutional networks: a comprehensive review. *Computational Social Networks*, 6(1):1–23, 2019. 3
- [55] Yuhao Zhang, Hang Jiang, Yasuhide Miura, Christopher D Manning, and Curtis P Langlotz. Contrastive learning of medical visual representations from paired images and text. *arXiv preprint arXiv:2010.00747*, 2020. 3
- [56] Yiwu Zhong, Jianwei Yang, Pengchuan Zhang, Chunyuan Li, Noel Codella, Liunian Harold Li, Luowei Zhou, Xiyang Dai, Lu Yuan, Yin Li, et al. Regionclip: Region-based language-image pretraining. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16793–16803, 2022. 3
- [57] Chong Zhou, Chen Change Loy, and Bo Dai. Extract free dense labels from CLIP. In *European Conference on Computer Vision*, volume 3, page 8, 2022. 3, 6, 8
- [58] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *International Journal of Computer Vision*, 130(9):2337–2348, 2022. 3
- [59] Yang Zou, Jongheon Jeong, Latha Pemula, Dongqing Zhang, and Onkar Dabeer. SPot-the-Difference self-supervised pre-training for anomaly detection and segmentation. In *Proceedings of the European Conference on Computer Vision*, 2022. 1, 3, 6, 12

Supplementary Material

WinCLIP: Zero-/Few-Shot Anomaly Classification and Segmentation

A. Experimental details

Compositional prompt ensemble. Figure 6 provides a detailed list of prompts we adopt to perform compositional prompt ensemble proposed in Section 4.1 of the main text. Recall that we consider two levels of prompts: *i.e.*, (a) state-level, and (b) template level. A complete prompt can be composed by replacing the token `[c]` in a template-level prompt with one of state-level prompt, either from the normal or anomaly states. Each of the state-level prompt takes an object-level label `[o]`. In our experiments, we use the object name words available for both MVTec-AD and VisA per dataset to replace `[o]`.

Data pre-processing. For CLIP-based models, including our proposed WinCLIP and WinCLIP+, we apply the data pre-processing pipeline given in OpenCLIP [16] for both MVTec-AD and VisA datasets to minimize potential train-test discrepancy. Specifically, it performs a channel-wise standardization with the pre-computed mean $[0.48145466, 0.4578275, 0.40821073]$ and standard deviation $[0.26862954, 0.26130258, 0.27577711]$ after normalizing each RGB image into $[0, 1]$, followed by a bicubic re-sizing based on the Pillow implementation. By default, we make the input resolution to be 240 for the shorter edge from the re-sizing, to be compatible with ViT-B/16+ in our experiments. This re-sizing policy also applies to other baseline models for fairer comparisons, although we keep the remaining parts of their original data pre-processing pipelines. In addition, similar policy can also be used in other backbones with input of different resolutions.

Evaluation metrics. Although the AUROC is a good metric for balanced dataset, it provides an inflated view of model performance in imbalanced dataset, especially in anomaly segmentation where the normal pixels dominate anomalies. This is also discussed by Zou et al. [59]. F_1 -max is computed from the precision and recall for the anomalous samples at the optimal threshold, which is a more straightforward metric to measure the upper bound of anomaly prediction performance across thresholds. Thus we acknowledge that the low-shot anomaly segmentation is still not solved since our best model only achieves $< 60\%$ F_1 -max for both MVTec-AD and VisA, even though WinCLIP+ achieves $> 95\%$ pixel-AUROC. In addition, our WinCLIP and WinCLIP+ outperform all the compared methods in terms of all these metrics on the setups, demonstrating the effectiveness of the proposed methods.

Other implementation details. (i) The ViT-B/16+ architecture [16], that we mainly adopt in our experiments, is a modification of ViT-B/16 [42] with (a) an increased dimension in both image ($768 \rightarrow 896$) and text ($512 \rightarrow 640$) embeddings, as well

(a) State-level (normal)

- `c := "[o]"`
- `c := "flawless [o]"`
- `c := "perfect [o]"`
- `c := "unblemished [o]"`
- `c := "[o] without flaw"`
- `c := "[o] without defect"`
- `c := "[o] without damage"`

(b) State-level (anomaly)

- `c := "damaged [o]"`
- `c := "[o] with flaw"`
- `c := "[o] with defect"`
- `c := "[o] with damage"`

(c) Template-level

- "a cropped photo of the `[c]`."
- "a close-up photo of a `[c]`."
- "a bright photo of a `[c]`."
- "a dark photo of the `[c]`."
- "a jpeg corrupted photo of a `[c]`."
- "a blurry photo of a `[c]`."
- "a photo of a `[c]`."
- "a photo of the `[c]`."
- "a photo of a small `[c]`."
- "a photo of the small `[c]`."
- "a photo of a large `[c]`."
- "a photo of the large `[c]`."
- "a photo of the `[c]` for visual inspection."
- "a photo of a `[c]` for visual inspection."
- "a photo of the `[c]` for anomaly detection."
- "a photo of a `[c]` for anomaly detection."

- (cont'd) "a blurry photo of the `[c]`."
- "a photo of a `[c]`."
- "a photo of the `[c]`."
- "a photo of a small `[c]`."
- "a photo of the small `[c]`."
- "a photo of a large `[c]`."
- "a photo of the large `[c]`."
- "a photo of the `[c]` for visual inspection."
- "a photo of a `[c]` for visual inspection."
- "a photo of the `[c]` for anomaly detection."
- "a photo of a `[c]` for anomaly detection."

Figure 6. Lists of multi-level prompts considered in this paper to construct compositional prompt ensemble.

MVTec-AD (few-shot)			Anomaly classification			Anomaly segmentation		
Setup	Method	Backbone	pAUROC	PRO	F_1 -max	AUROC	AUPR	F_1 -max
1-shot	PatchCore [31]	WRN-50-2	83.4 \pm 3.0	92.2 \pm 1.5	90.5 \pm 1.5	92.0 \pm 1.0	79.7 \pm 2.0	50.4 \pm 2.1
	PatchCore (hidden)	ViT-B/16+	79.9 \pm 4.8	88.6 \pm 2.7	88.7 \pm 1.1	91.8 \pm 1.0	74.1 \pm 2.3	47.6 \pm 2.6
	PatchCore (last)		83.3 \pm 3.8	90.7 \pm 2.1	89.8 \pm 1.4	92.3 \pm 0.9	74.5 \pm 2.2	47.7 \pm 2.9
	WinCLIP+ (ours)		92.7\pm1.9	96.7\pm0.7	93.5\pm1.0	95.2\pm0.5	87.1\pm1.2	55.9\pm2.7
2-shot	PatchCore [31]	WRN-50-2	86.3 \pm 3.3	93.8 \pm 1.7	92.0 \pm 1.5	93.3 \pm 0.6	82.3 \pm 1.3	53.0 \pm 1.7
	PatchCore (hidden)	ViT-B/16+	84.1 \pm 2.9	90.7 \pm 1.9	90.2 \pm 1.2	93.5 \pm 0.7	77.9 \pm 1.8	51.4 \pm 2.1
	PatchCore (last)		86.5 \pm 2.5	92.3 \pm 1.4	91.1 \pm 1.6	93.1 \pm 0.9	76.8 \pm 2.0	49.8 \pm 2.2
	WinCLIP+ (ours)		94.0\pm1.7	97.0\pm0.7	94.0\pm1.0	96.0\pm0.3	88.4\pm0.9	58.4\pm1.7
4-shot	PatchCore [31]	WRN-50-2	88.8 \pm 2.6	94.5 \pm 1.5	92.6 \pm 1.6	94.3 \pm 0.5	84.3 \pm 1.6	55.0 \pm 1.9
	PatchCore (hidden)	ViT-B/16+	87.5 \pm 3.1	92.5 \pm 1.8	91.7 \pm 1.4	94.7 \pm 0.6	81.2 \pm 1.6	54.4 \pm 1.9
	PatchCore (last)		89.9 \pm 2.2	94.2 \pm 1.4	92.7 \pm 1.1	94.0 \pm 0.6	78.9 \pm 1.6	52.2 \pm 1.5
	WinCLIP+ (ours)		94.8\pm1.5	97.5\pm0.7	94.2\pm0.9	96.2\pm0.3	89.0\pm0.8	59.5\pm1.8

Table 8. Comparison of few-shot performances on MVTec-AD. We report the mean and standard deviation over 5 random seeds for each measurement. Bold indicates the best performance.

MVTec-AD (zero-shot)			
Model	Size	AC	AS
RN50	224 ²	79.8	65.6
RN101	224 ²	79.2	62.4
RN50x4	288 ²	81.9	71.3
RN50x16	384 ²	82.3	65.3
ViT-B/16	224 ²	86.1	71.1
ViT-B/16+	240 ²	90.8	85.1
ViT-L/14	224 ²	86.1	64.4

Table 9. Comparison of WinCLIP performance in AUROC (for AC) and pAUROC (for AS) on zero-shot MVTec-AD, across different CLIP backbone architectures.

as in (b) the input resolution ($224^2 \rightarrow 240^2$; $196 \rightarrow 225$ tokens); (ii) We note that CLIP models require the square-shaped resolution, *e.g.*, 240^2 for ViT-B/16+, to be compatible with the attention layers inside. Although the MVTec-AD benchmark already consists of square images, most of images in the VisA benchmark are non-squared (*e.g.*, 1500×1000) and simply taking a crop can affect the anomaly status of the given images. In this respect, to enable CLIP-based models properly handle non-squared images in our experiments, we perform a simple “image tiling” scheme. Specifically, for such non-squared images, we first extract multiple overlapping (squared) “tiles” of size the shorter edge L_s , by taking a sliding window across the longer edge. Then we average the predictions from the tiles to get the final (either in image- and pixel-level) prediction. The stride for the sliding is set to $0.8 \cdot L_s$ at most, *i.e.*, the tiles have overlaps with its neighbors at least in $0.2 \cdot L_s$; (iii) In addition, for the baseline results, we use our re-implementation of SPADE [7] and PaDiM [8], and adopt the official implementation of PatchCore⁴ in our experiments.

B. Additional results on ablation study

Comparison with CLIP-based PatchCore: PatchCore [31], a current state-of-the-art considered in our experiments, is originally based on the internal features of convolutional network: *e.g.*, WideResNet-50-2 (WRN-50-2) [52] pre-trained on ImageNet. In Table 8, we test whether PatchCore can further benefit from the CLIP-based backbone that our WinCLIP+ is based on. Specifically, we additionally consider two variants of PatchCore that take the patch-token features of CLIP-based ViT-B/16+ backbone, one from (a) the 6th- and 9th-layer of ViT (which corresponds to block2 and block3 in ResNet-like models as considered by [31]; “hidden”), and the other one from (b) the last layer of ViT (“last”). Overall, we have the following observations. First, in case of the ViT-B/16+ backbone, PatchCore performs better with the last layer, which is in contrast to the cases of convolutional backbones. Second, compared to the original PatchCore, the CLIP-based variants achieve no better performances. Third, WinCLIP+ significantly outperforms “PatchCore (last)” where our WinCLIP+ also utilizes the last patch-token features, namely as referred as \mathbf{F}^p (Section 4.3 of the main text). The results confirm the effectiveness of (a) our simple association-based module over a more sophisticated PatchCore⁵ in ViT, and (b) the WinCLIP features \mathbf{F}^w .

Effect of different CLIP backbones: Table 9, on the other hand, explores the effect of different CLIP architectures to the WinCLIP zero-shot performance. Specifically, on zero-shot setups, we compare AUROC (and pixel-AUROC) from WinCLIP in AC (and AS) testing over the CLIP pre-trained models available at OpenCLIP,⁶ including our default choice of ViT-B/16+. To apply WinCLIP for ResNet-based backbones, we notice that the CLIP implementation of ResNet architectures incorporates an attention layer to perform the feature pooling, namely as *attention pooling*, similar to ViT-based architectures. In this respect, for the CLIP-ResNet models, we apply our window-based inference to perform zero-shot AS from the convolutional feature map before the attention pooling, in the same way of applying WinCLIP for ViTs. Here, we remark that the effective patch size of each pixel on the last feature map (before the pooling) of ResNet-based models is designed to be 32 (the downsampling

⁴<https://github.com/amazon-science/patchcore-inspection>

⁵Technically, PatchCore incorporates several techniques upon a patch-level memory scheme, *e.g.*, local patch aggregation, clustering and score re-weighting.

⁶https://github.com/mlfoundations/open_clip

rate), which is larger than those of ViTs we test, *e.g.*, of 16. Overall, we observe that ViT-based models generally show better performance compared to ResNets, in both AC and AS. The particular gap in AS is possibly due to the bigger patch sizes in ResNets, which can result in more blurry outputs. Still, we observe the performance benefits from larger models or resolutions in both types of architecture.

C. Additional qualitative results

In Figure 7-10, we provide further qualitative results obtained from our (zero-shot) WinCLIP and (few-shot) WinCLIP+ for anomaly segmentation, both in MVTec-AD and VisA considered in our experiments. Specifically, we report MVTec-AD results in Figure 7 and 8, and VisA results in Figure 9 and 10.

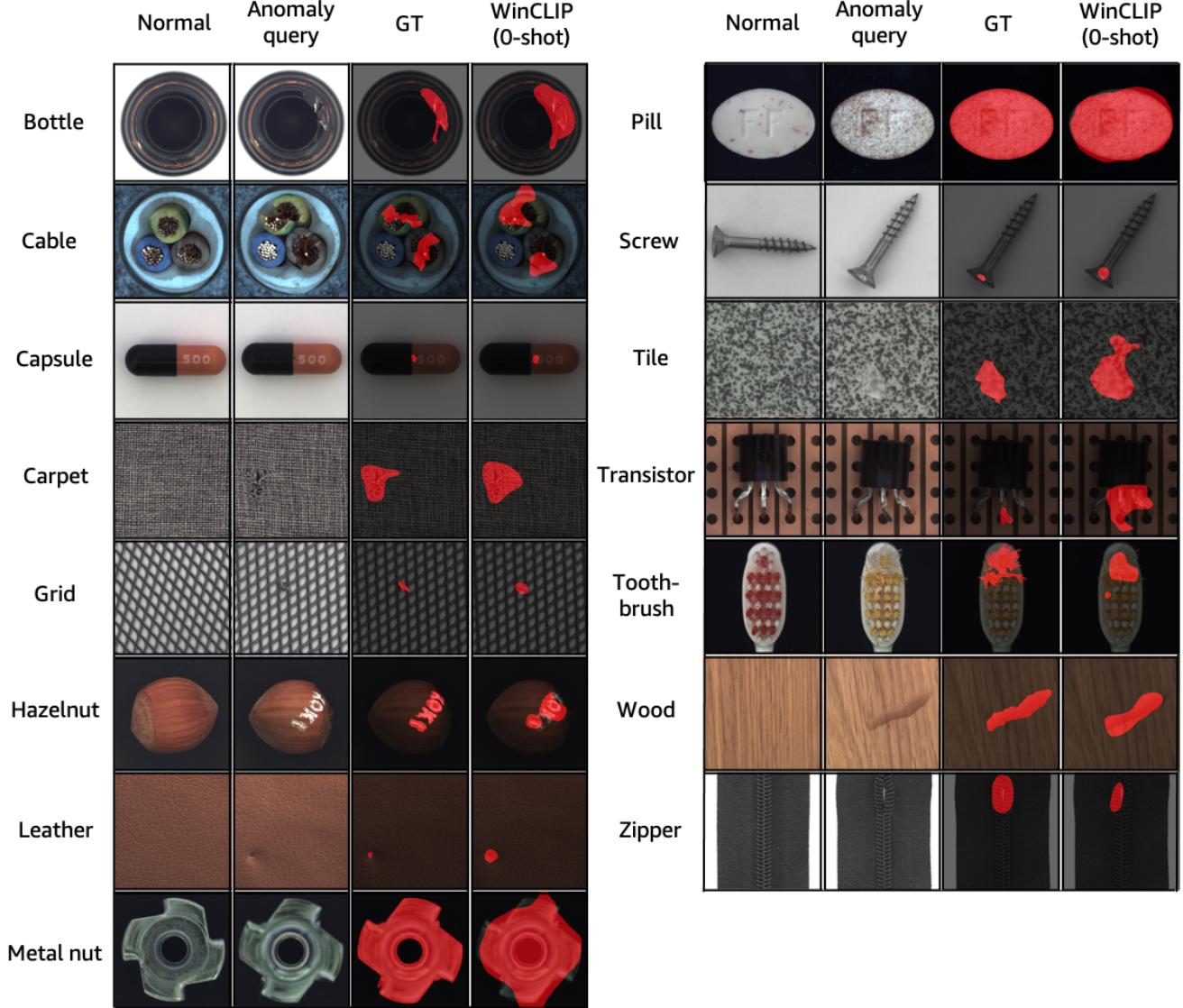


Figure 7. Additional qualitative results from WinCLIP (0-shot), tested on MVTec-AD.

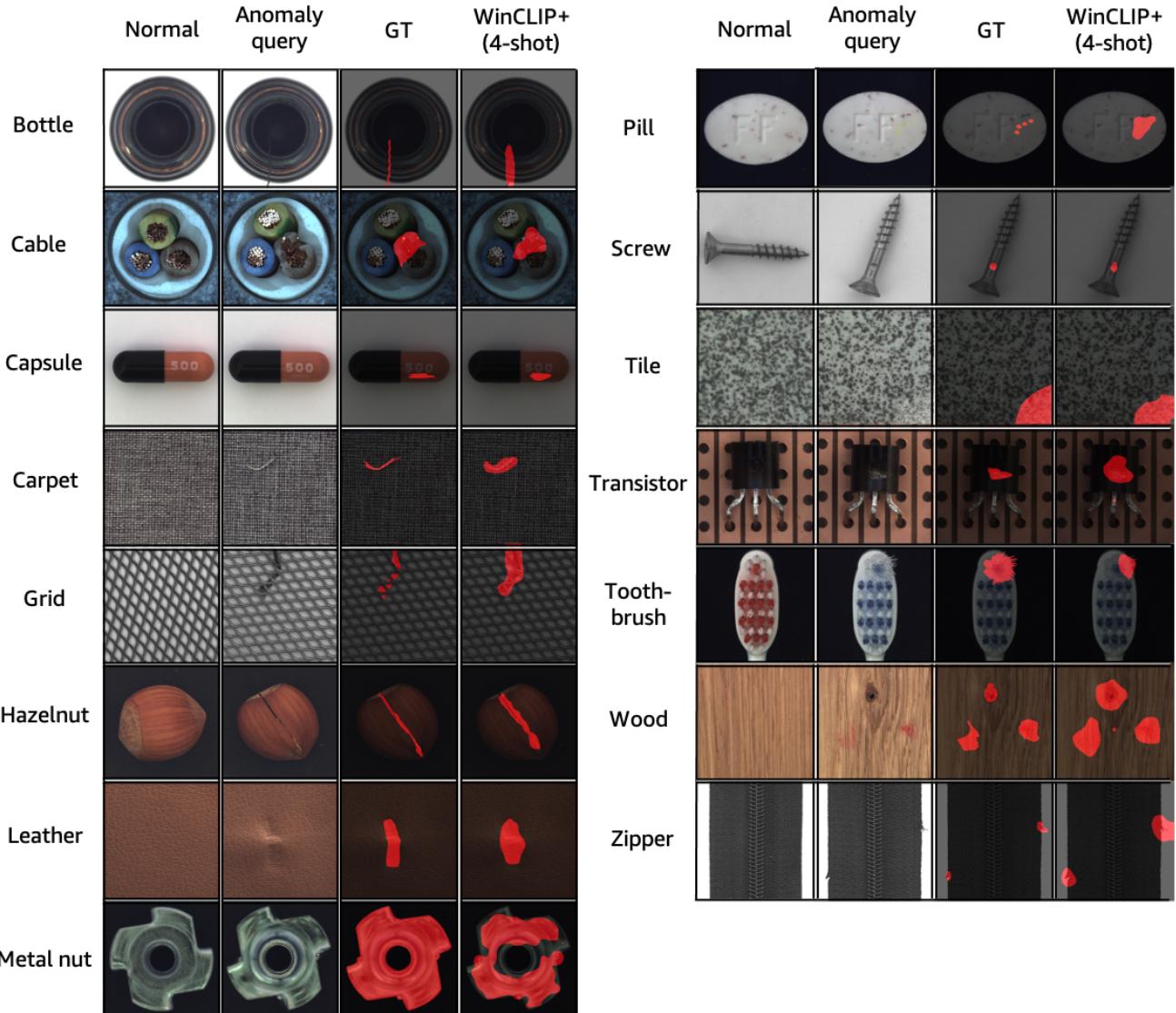


Figure 8. Additional qualitative results from few-shot WinCLIP+ (4-shot), tested on MVTec-AD.

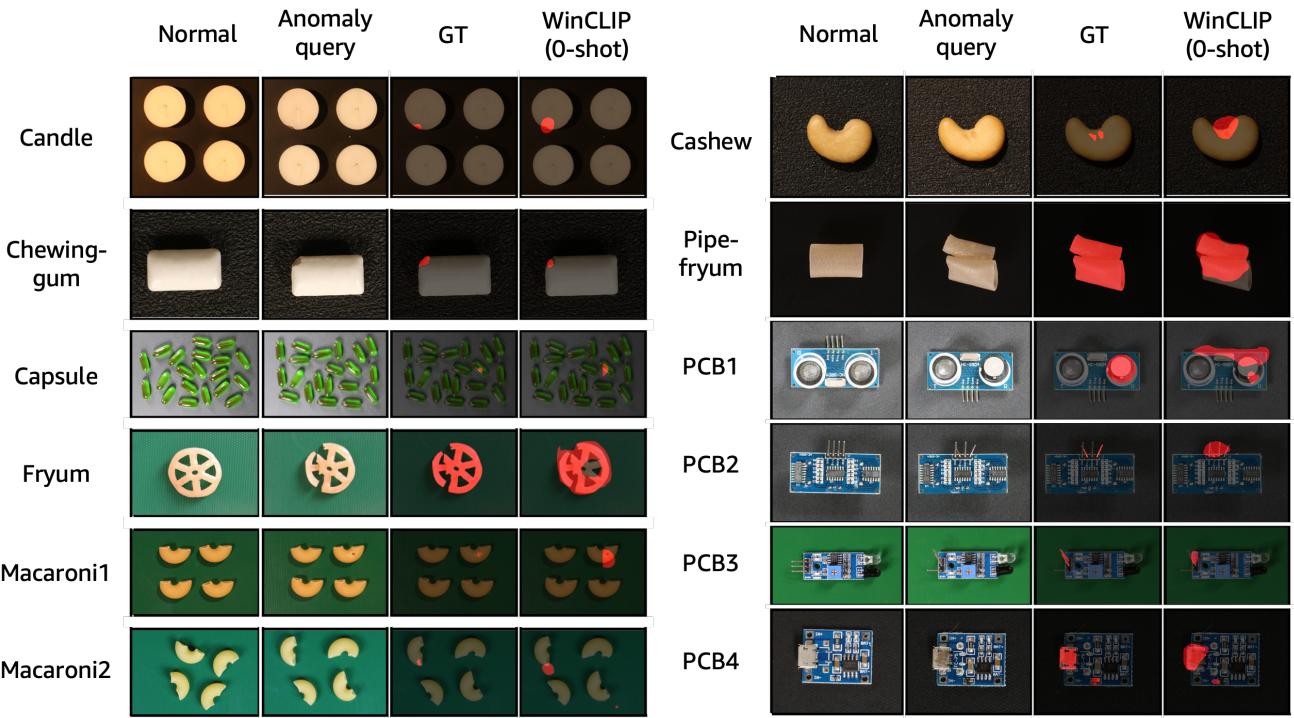


Figure 9. Additional qualitative results from WinCLIP (0-shot), tested on VisA.

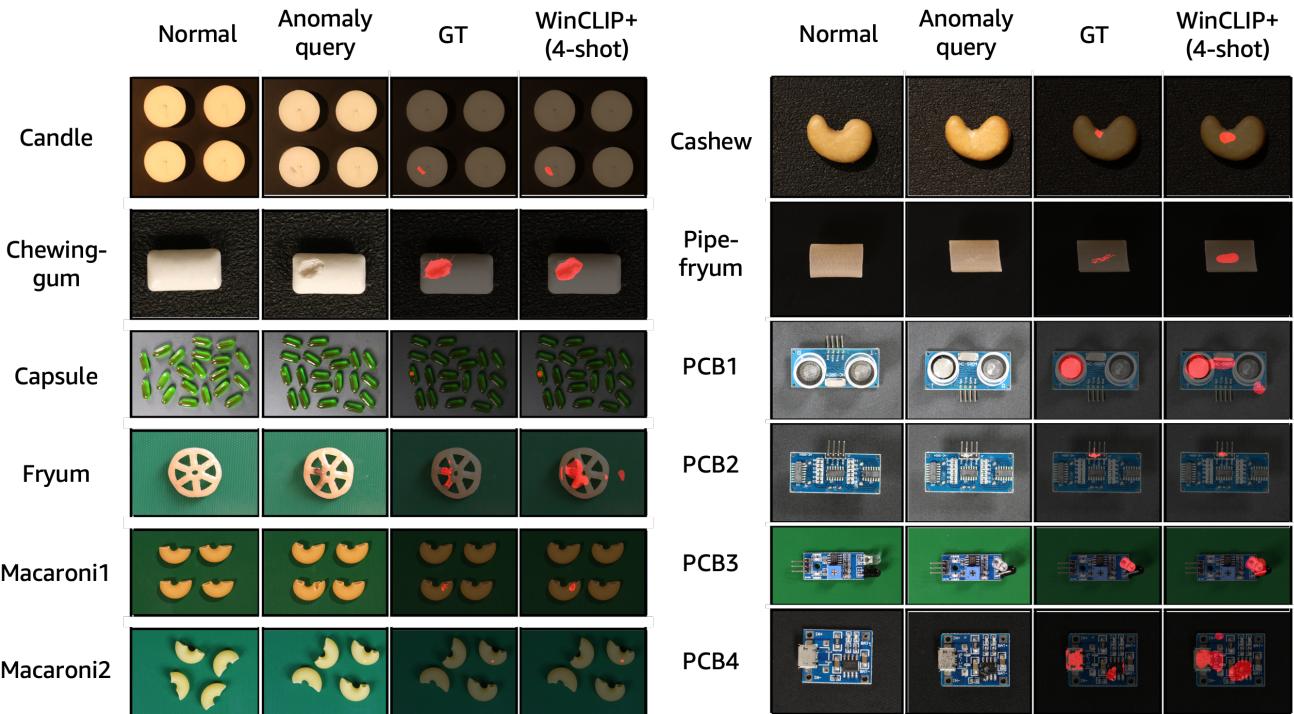


Figure 10. Additional qualitative results from few-shot WinCLIP+ (4-shot), tested on VisA.

Failure cases. We present some failure examples from both MVTec-AD and VisA for language driven zero-shot WinCLIP in Figure 11. Note that the normal images are shown just for better illustration and are not used in model prediction. The first major factor causing the failure is the logical anomaly [2] illustrated in Figure 11(a), e.g., misplaced axis in cable, missing text on capsule, missing capacitor in PCB1 and bent component in PCB3. Such type of anomalies need to be clarified by normal reference images while language might be not sufficient. The issues are alleviated by our few-normal-shot WinCLIP+. The second major factor refers to tiny defect illustrated in Figure 11(b), such as the ones in carpet, wood, capsules, macaroni1. We conjecture that spatial features with more local details might improve these cases, which is left for future exploration. The third major factor is the irrelevant deviation from normality that are not defects of interests illustrated in Figure 11(c), e.g., the tiny red/white dots in pill/hazelnut, extra ingredient on cashew, designed holes and acceptable scratches in PCB2. We hypothesize that more clarification on these deviation and a pre-trained model with better understanding on these states might alleviate the problem. Lastly, although WinCLIP can roughly localize anomalies such as the cases in bottle, tile, PCB4 and fryum, it makes some errors around the true positives, illustrated in Figure 11(d). However, we argue this is minor as the rough anomaly localization is sufficient to explain where the defects are for visual inspection.

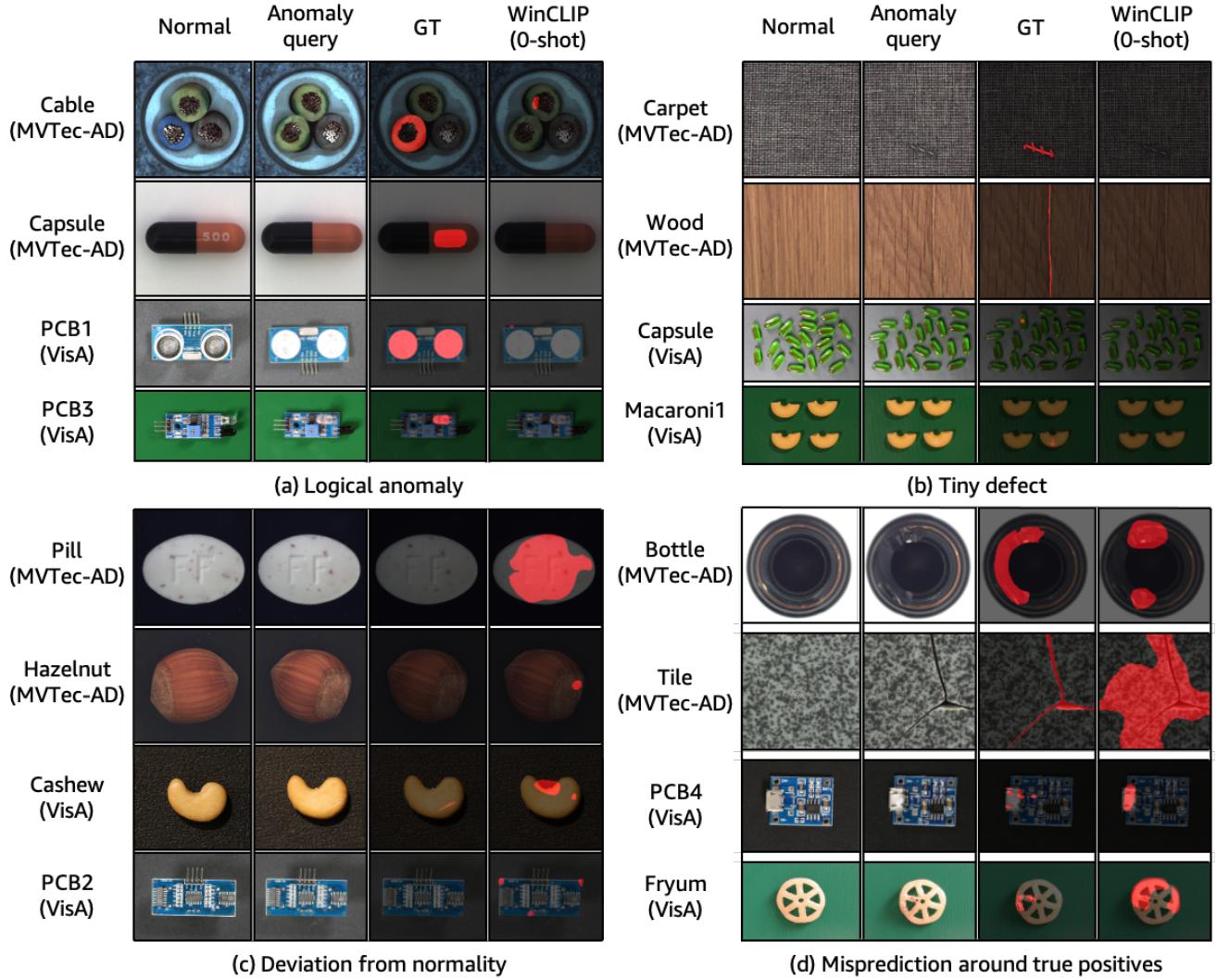


Figure 11. Curated illustrations of failure cases from zero-shot WinCLIP.

D. Detailed quantitative results

In this section, we report the detailed, subset-level performance values for the evaluation metrics provided in Table 1 and 4 of the main text. Specifically, we report MVTec-AD results in Table 10-15 and VisA results in Table 16-21.

MVTec-AD (AC)	$K = 0$	$K = 1$				$K = 2$				$K = 4$			
		AUROC	WinCLIP	SPADE	PaDiM	PatchCore	WinCLIP+	SPADE	PaDiM	PatchCore	WinCLIP+	SPADE	PaDiM
Bottle	99.2±0.0	98.7±0.6	97.4±0.7	99.4±0.4	98.2±0.9	99.5±0.1	98.5±1.0	99.2±0.3	99.3±0.3	99.5±0.2	98.8±0.2	99.2±0.3	99.3±0.4
Cable	86.5±0.0	71.2±3.3	57.7±4.6	88.8±4.2	88.9±1.9	76.2±5.2	62.3±5.9	91.0±2.7	88.4±0.7	83.4±3.1	70.0±6.1	91.0±2.7	90.9±0.9
Capsule	72.9±0.0	70.2±3.0	57.7±7.3	67.8±2.9	72.3±6.8	70.9±6.1	64.3±3.0	72.8±7.0	77.3±8.8	78.9±5.5	65.2±2.5	72.8±7.0	82.3±8.9
Carpet	100.0±0.0	98.1±0.2	96.6±1.0	95.3±0.8	99.8±0.3	98.3±0.4	97.8±0.5	96.6±0.5	99.8±0.3	98.6±0.2	97.9±0.4	96.6±0.5	100.0±0.0
Grid	98.8±0.0	40.0±6.8	54.2±6.7	63.6±10.3	99.5±0.3	41.3±3.6	67.2±4.2	67.7±8.3	99.4±0.2	44.6±6.6	68.1±3.8	67.7±8.3	99.6±0.1
Hazelnut	93.9±0.0	95.8±1.3	88.3±2.6	88.3±2.7	97.5±1.4	96.2±2.1	90.8±0.8	93.2±3.8	98.3±0.7	98.4±1.3	91.9±1.2	93.2±3.8	98.4±0.4
Leather	100.0±0.0	100.0±0.0	97.5±0.7	97.3±0.7	99.9±0.0	100.0±0.0	97.5±0.9	97.9±0.7	99.9±0.0	100.0±0.0	98.5±0.2	97.9±0.7	100.0±0.0
Metal nut	97.1±0.0	71.0±2.2	53.0±3.8	73.4±2.9	98.7±0.8	77.0±7.9	54.8±3.8	77.7±8.5	99.4±0.2	77.8±5.7	60.7±5.2	77.7±8.5	99.5±0.2
Pill	79.1±0.0	86.5±3.1	61.3±3.8	81.9±2.8	91.2±2.1	84.8±0.9	59.1±6.4	82.9±2.9	92.3±0.7	86.7±0.3	54.9±2.7	82.9±2.9	92.8±1.0
Screw	83.3±0.0	46.7±2.5	55.0±2.5	44.4±4.6	86.4±0.9	46.6±2.2	54.0±4.4	49.0±3.8	86.0±2.1	50.5±5.4	50.0±4.1	49.0±3.8	87.9±1.2
Tile	100.0±0.0	99.9±0.1	92.2±2.2	99.0±0.9	99.9±0.0	99.9±0.1	93.3±1.1	98.5±1.0	99.9±0.2	100.0±0.0	93.1±0.6	98.5±1.0	99.9±0.1
Toothbrush	87.5±0.0	71.7±2.6	82.5±1.2	83.3±3.8	92.2±4.9	78.6±3.2	87.6±4.2	85.9±3.5	97.5±1.6	78.8±5.2	89.2±2.5	85.9±3.5	96.7±2.6
Transistor	88.0±0.0	77.2±2.0	73.3±6.0	78.1±6.9	83.4±3.8	81.3±3.7	72.8±6.3	90.0±4.3	85.3±1.7	81.4±2.1	82.4±6.5	90.0±4.3	85.7±2.5
Wood	99.4±0.0	98.8±0.3	96.1±1.2	97.8±0.3	99.9±0.1	99.2±0.4	96.9±0.5	98.3±0.6	99.9±0.1	98.9±0.6	97.0±0.2	98.3±0.6	99.8±0.3
Zipper	91.5±0.0	89.3±1.9	85.8±2.7	92.3±0.5	88.8±5.9	93.3±2.9	86.3±2.6	94.0±2.1	94.0±1.4	95.1±1.3	88.3±2.0	94.0±2.1	94.5±0.5
Mean	91.8±0.0	81.0±2.0	76.6±3.1	83.4±3.0	93.1±2.0	82.9±2.6	78.9±3.1	86.3±3.3	94.4±1.3	84.8±2.5	80.4±2.5	88.8±2.6	95.2±1.3

Table 10. Comparison of anomaly classification (AC) performance in terms of class-wise AUROC on MVTec-AD. We report the mean and standard deviation over 5 random seeds for each measurement.

MVTec-AD (AC)	$K = 0$	$K = 1$				$K = 2$				$K = 4$			
		AUPR	WinCLIP	SPADE	PaDiM	PatchCore	WinCLIP+	SPADE	PaDiM	PatchCore	WinCLIP+	SPADE	PaDiM
Bottle	99.8±0.0	99.6±0.1	99.2±0.2	99.8±0.1	99.4±0.3	99.8±0.0	99.6±0.3	99.8±0.1	99.8±0.1	99.9±0.0	99.7±0.0	99.8±0.1	99.8±0.1
Cable	91.2±0.0	79.6±2.3	64.9±3.8	93.8±2.2	93.2±1.1	84.5±3.1	69.6±6.6	95.1±1.3	92.9±0.6	88.8±1.9	76.1±5.6	97.1±0.7	94.4±0.3
Capsule	91.5±0.0	91.2±0.9	86.9±2.2	89.4±2.0	91.6±2.7	91.6±2.1	88.4±0.8	91.0±2.9	93.3±3.6	94.4±1.9	87.8±0.8	94.9±1.1	95.1±3.3
Carpet	100.0±0.0	99.4±0.0	99.0±0.2	98.7±0.2	99.9±0.1	99.5±0.1	99.4±0.1	99.0±0.1	99.9±0.1	99.6±0.1	99.4±0.1	98.8±0.2	100.0±0.0
Grid	99.6±0.0	66.9±2.1	75.0±3.3	81.1±4.9	99.9±0.1	68.3±2.1	82.5±2.3	84.1±4.0	99.8±0.1	68.8±4.2	83.0±1.8	86.4±4.0	99.9±0.0
Hazelnut	96.9±0.0	97.9±0.6	93.3±1.7	92.9±2.2	98.6±0.7	98.0±1.1	94.1±0.5	96.0±2.0	99.1±0.4	99.1±0.7	94.8±0.6	97.0±1.2	99.1±0.2
Leather	100.0±0.0	100.0±0.0	99.2±0.2	99.1±0.2	100.0±0.0	100.0±0.0	99.2±0.3	99.3±0.2	100.0±0.0	100.0±0.0	99.6±0.1	99.6±0.1	100.0±0.0
Metal nut	99.3±0.0	91.7±0.8	82.0±2.7	91.0±1.1	99.7±0.2	93.7±2.4	82.2±1.4	92.3±4.0	99.9±0.0	94.1±1.8	85.5±1.7	97.0±2.6	99.9±0.1
Pill	95.7±0.0	97.0±0.8	88.3±1.3	96.5±0.6	98.3±0.5	96.5±0.4	87.9±2.6	96.6±0.7	98.6±0.1	97.0±0.2	87.0±1.2	96.9±0.4	98.6±0.2
Screw	93.1±0.0	71.3±1.8	78.1±1.0	71.4±2.3	94.2±0.6	71.0±1.4	77.3±1.3	72.9±3.4	94.1±1.5	73.7±2.4	75.7±2.8	71.8±1.9	94.9±0.8
Tile	100.0±0.0	100.0±0.0	97.2±0.7	96.9±0.3	100.0±0.0	100.0±0.0	97.6±0.4	99.4±0.4	100.0±0.1	100.0±0.0	97.6±0.2	99.6±0.1	100.0±0.0
Toothbrush	95.6±0.0	88.3±0.6	93.7±0.5	93.5±1.4	96.7±2.0	90.8±1.3	95.2±1.6	94.1±1.4	99.0±0.6	91.3±2.6	95.8±0.7	94.8±0.7	98.7±1.1
Transistor	87.1±0.0	76.2±1.7	66.2±2.5	77.7±5.5	79.0±4.0	81.6±3.4	69.0±6.5	89.3±3.9	80.7±2.3	80.3±2.6	77.6±8.4	84.5±9.0	80.7±3.2
Wood	99.8±0.0	99.6±0.1	98.8±0.3	99.3±0.1	100.0±0.0	99.7±0.1	99.0±0.1	99.5±0.2	100.0±0.0	99.7±0.2	99.1±0.0	99.5±0.2	99.9±0.1
Zipper	97.5±0.0	96.9±0.5	95.5±0.9	97.2±0.3	96.8±1.8	98.2±0.8	95.4±1.0	97.8±1.0	98.3±0.4	98.6±0.4	96.2±0.8	99.1±0.7	98.5±0.2
Mean	96.5±0.0	90.6±0.8	88.1±1.7	92.2±1.5	96.5±0.9	91.7±1.2	89.3±1.7	93.8±1.7	97.0±0.7	92.5±1.2	90.5±1.6	94.5±1.5	97.3±0.6

Table 11. Comparison of anomaly classification (AC) performance in terms of class-wise AUPR on MVTec-AD. We report the mean and standard deviation over 5 random seeds for each measurement.

MVTec-AD (AC)	$K = 0$	$K = 1$				$K = 2$				$K = 4$			
		F_1 -max	WinCLIP	SPADE	PaDiM	PatchCore	WinCLIP+	SPADE	PaDiM	PatchCore	WinCLIP+	SPADE	PaDiM
Bottle	97.6±0.0	97.8±0.8	96.3±1.2	98.3±0.6	96.5±1.3	98.7±0.4	97.1±1.1	97.5±0.6	97.7±0.7	98.6±0.3	97.9±0.4	97.9±0.8	97.8±0.6
Cable	84.5±0.0	79.6±2.3	77.2±1.1	85.2±3.6	86.1±1.3	80.4±1.7	78.7±1.2	86.1±2.4	85.2±0.7	83.8±2.5	81.1±1.1	91.3±1.0	87.2±0.6
Capsule	91.4±0.0	92.0±0.6	91.0±0.2	92.0±1.0	91.6±0.7	92.1±0.4	92.1±0.9	93.6±0.6	92.1±0.7	92.7±0.3	92.8±0.9	94.3±0.3	92.5±0.5
Carpet	99.4±0.0	96.5±0.2	95.1±0.5	94.9±0.5	99.2±0.8	96.6±0.3	96.5±0.4	95.3±0.5	99.3±0.7	96.9±0.3	96.6±0.3	94.3±0.8	99.9±0.2
Grid	98.2±0.0	84.5±0.3	84.5±0.3	86.2±1.1	98.9±0.4	84.8±0.3	85.3±0.9	86.9±2.3	99.1±0.0	84.8±0.5	85.0±0.5	87.5±2.0	99.1±0.0
Hazelnut	89.7±0.0	92.4±1.3	87.4±1.6	87.0±1.4	94.7±2.3	93.2±2.8	89.3±1.1	91.0±3.7	95.6±1.6	95.9±2.0	90.0±1.7	92.8±1.2	96.2±1.0
Leather	100.0±0.0	99.9±0.2	96.2±0.9	95.9±0.7	99.5±0.0	100.0±0.0	96.6±1.3	95.7±0.9	99.7±0.2	100.0±0.0	97.9±0.2	97.5±0.7	99.8±0.2
Metal nut	96.3±0.0	90.1±0.6	90.1±0.3	91.4±1.2	97.7±1.0	90.5±1.1	90.0±0.3	91.9±0.9	98.4±0.5	90.6±0.9	90.3±0.4	93.6±1.4	98.5±0.6
Pill	91.6±0.0	93.5±0.4	91.6±0.0	91.9±0.3	93.8±0.7	93.4±0.3	91.7±0.1	92.0±0.3	94.3±0.4	93.6±0.5	91.7±0.1	92.1±0.3	94.1±0.4
Screw	87.4±0.0	85.3±0.0	85.6±0.3	85.8±0.2	88.5±0.3	85.6±0.2	85.5±0.1	85.7±0.2	89.0±0.6	85.8±0.7	85.7±0.3	86.8±0.6	89.6±0.7
Tile	99.4±0.0	99.2±0.5	90.7±2.0	97.5±1.2	98.9±0.2	99.2±0.3	91.5±1.3	96.9±1.5	99.2±0.3	99.4±0.0	91.0±0.9	97.5±0.4	99.2±0.3
Toothbrush	87.9±0.0	85.6±1.3	85.4±1.3	88.9±2.2	94.1±1.9	87.1±1.4	90.1±3.1	90.8±1.6	96.7±1.8	86.5±1.8	90.6±2.1	92.6±2.2	96.8±2.3
Transistor	79.5±0.0	70.3±1.7	66.4±4.7	70.6±7.2	75.1±3.1	73.4±3.4	65.9±3.7	85.3±6.1	75.9±2.4	72.3±2.7	74.8±7.7	78.3±11.5	76.6±2.8
Wood	98.3±0.0	97.0±0.8	94.3±1.1	96.1±0.3	99.4±0.3	97.7±1.0	94.9±0.3	96.6±0.9	99.5±0.4	97.6±0.7	94.8±0.5	96.5±0.9	99.2±0.9
Zipper	92.9±0.0	91.0±0.9	91.2±0.8	95.3±0.5	92.1±2.5	93.5±1.8	92.2±0.6	95.4±0.5	94.4±0.3	94.7±0.8	92.5±0.8	96.5±0.3	94.7±0.4
Mean	92.9±0.0	90.3±0.8	88.2±1.1	90.5±1.5	93.7±1.1	91.1±1.0	89.2±1.1	92.0±1.5	94.4±0.8	91.5±0.9	90.2±1.2	92.6±1.6	94.7±0.8

Table 12. Comparison of anomaly classification (AC) performance in terms of class-wise F_1 -max on MVTec-AD. We report the mean and standard deviation over 5 random seeds for each measurement.

MVTec-AD (AS)	$K = 0$	$K = 1$				$K = 2$				$K = 4$			
pAUROC	WinCLIP	SPADE	PaDiM	PatchCore	WinCLIP+	SPADE	PaDiM	PatchCore	WinCLIP+	SPADE	PaDiM	PatchCore	WinCLIP+
Bottle	89.5 \pm 0.0	95.3 \pm 0.2	96.1 \pm 0.5	97.9 \pm 0.1	97.5 \pm 0.2	95.7 \pm 0.2	96.9 \pm 0.1	98.1 \pm 0.0	97.7 \pm 0.1	96.1 \pm 0.0	97.1 \pm 0.1	98.2 \pm 0.0	97.8 \pm 0.0
Cable	77.0 \pm 0.0	86.4 \pm 0.2	88.4 \pm 1.2	95.5 \pm 0.8	93.8 \pm 0.6	87.4 \pm 0.3	90.0 \pm 0.8	96.4 \pm 0.3	94.3 \pm 0.4	88.2 \pm 0.2	92.1 \pm 0.4	97.5 \pm 0.3	94.9 \pm 0.1
Capsule	86.9 \pm 0.0	96.3 \pm 0.2	94.5 \pm 0.6	95.6 \pm 0.4	94.6 \pm 0.8	96.7 \pm 0.1	95.2 \pm 0.5	96.5 \pm 0.4	96.4 \pm 0.3	97.0 \pm 0.2	96.2 \pm 0.4	96.8 \pm 0.6	96.2 \pm 0.5
Carpet	95.4 \pm 0.0	98.2 \pm 0.0	97.8 \pm 0.2	98.4 \pm 0.1	99.4 \pm 0.0	98.3 \pm 0.0	98.2 \pm 0.0	98.5 \pm 0.1	99.3 \pm 0.0	98.4 \pm 0.0	98.4 \pm 0.0	98.6 \pm 0.1	99.3 \pm 0.0
Grid	82.2 \pm 0.0	80.7 \pm 1.3	70.2 \pm 2.8	58.8 \pm 4.9	96.8 \pm 1.0	83.5 \pm 1.0	70.8 \pm 2.0	62.6 \pm 3.2	97.7 \pm 0.8	87.2 \pm 1.1	77.0 \pm 1.8	69.4 \pm 1.3	98.0 \pm 0.2
Hazelnut	94.3 \pm 0.0	97.2 \pm 0.1	95.4 \pm 0.6	95.8 \pm 0.6	98.5 \pm 0.2	97.6 \pm 0.1	96.8 \pm 0.3	96.3 \pm 0.6	98.7 \pm 0.1	97.7 \pm 0.1	97.2 \pm 0.2	97.6 \pm 0.1	98.8 \pm 0.0
Leather	96.7 \pm 0.0	99.1 \pm 0.0	98.5 \pm 0.1	98.8 \pm 0.2	99.3 \pm 0.0	99.1 \pm 0.0	98.7 \pm 0.1	99.0 \pm 0.1	99.3 \pm 0.0	99.1 \pm 0.0	98.8 \pm 0.0	99.1 \pm 0.0	99.3 \pm 0.0
Metal nut	61.0 \pm 0.0	83.8 \pm 0.7	74.6 \pm 1.1	89.3 \pm 1.4	90.0 \pm 0.6	85.8 \pm 1.1	80.3 \pm 2.1	94.6 \pm 1.4	91.4 \pm 0.4	87.1 \pm 0.7	82.7 \pm 3.9	95.9 \pm 1.8	92.9 \pm 0.4
Pill	80.0 \pm 0.0	89.4 \pm 0.4	84.8 \pm 1.0	93.1 \pm 1.1	96.4 \pm 0.3	89.9 \pm 0.2	87.3 \pm 0.7	94.2 \pm 0.3	97.0 \pm 0.2	90.7 \pm 0.2	88.9 \pm 0.5	94.8 \pm 0.4	97.1 \pm 0.0
Screw	89.6 \pm 0.0	94.8 \pm 0.2	83.3 \pm 0.7	89.6 \pm 0.5	94.5 \pm 0.4	95.6 \pm 0.4	89.8 \pm 0.8	90.0 \pm 0.7	95.2 \pm 0.3	96.4 \pm 0.4	90.8 \pm 0.2	91.3 \pm 1.0	96.0 \pm 0.5
Tile	77.6 \pm 0.0	91.7 \pm 0.3	84.1 \pm 1.1	94.1 \pm 0.5	96.3 \pm 0.2	92.0 \pm 0.1	87.7 \pm 0.2	94.4 \pm 0.2	96.5 \pm 0.1	92.2 \pm 0.1	88.9 \pm 0.3	94.6 \pm 0.1	96.6 \pm 0.1
Toothbrush	86.9 \pm 0.0	94.6 \pm 0.6	97.3 \pm 0.3	97.3 \pm 0.4	97.8 \pm 0.1	96.2 \pm 0.3	97.7 \pm 0.3	97.5 \pm 0.2	98.1 \pm 0.1	97.0 \pm 0.6	98.4 \pm 0.2	98.4 \pm 0.5	
Transistor	74.7 \pm 0.0	71.4 \pm 1.3	90.2 \pm 2.8	84.9 \pm 2.7	85.0 \pm 1.8	72.8 \pm 0.9	92.3 \pm 2.1	89.6 \pm 0.9	88.3 \pm 1.0	73.4 \pm 0.7	94.0 \pm 2.7	90.7 \pm 1.4	88.5 \pm 1.2
Wood	93.4 \pm 0.0	93.4 \pm 0.1	90.7 \pm 0.4	92.7 \pm 0.9	94.6 \pm 1.0	93.8 \pm 0.1	91.9 \pm 0.1	93.2 \pm 0.7	95.3 \pm 0.4	93.9 \pm 0.1	92.2 \pm 0.1	93.5 \pm 0.3	95.4 \pm 0.2
Zipper	91.6 \pm 0.0	94.9 \pm 0.3	93.9 \pm 0.8	97.4 \pm 0.4	93.9 \pm 0.8	95.8 \pm 0.2	95.4 \pm 0.3	98.0 \pm 0.1	94.1 \pm 0.7	96.2 \pm 0.1	96.1 \pm 0.2	98.1 \pm 0.1	94.2 \pm 0.4
Mean	85.1\pm0.0	91.2 \pm 0.4	89.3 \pm 0.9	92.0 \pm 1.0	95.2\pm0.5	92.0 \pm 0.3	91.3 \pm 0.7	93.3 \pm 0.6	96.0\pm0.3	92.7 \pm 0.3	92.6 \pm 0.7	94.3 \pm 0.5	96.2\pm0.3

Table 13. Comparison of anomaly segmentation (AS) performance in terms of class-wise pixel-AUROC on MVTec-AD. We report the mean and standard deviation over 5 random seeds for each measurement.

MVTec-AD (AS)	$K = 0$	$K = 1$				$K = 2$				$K = 4$			
PRO	WinCLIP	SPADE	PaDiM	PatchCore	WinCLIP+	SPADE	PaDiM	PatchCore	WinCLIP+	SPADE	PaDiM	PatchCore	WinCLIP+
Bottle	76.4 \pm 0.0	91.1 \pm 0.4	89.8 \pm 0.8	93.5 \pm 0.3	91.2 \pm 0.4	91.8 \pm 0.5	91.7 \pm 0.2	93.9 \pm 0.3	91.8 \pm 0.3	92.5 \pm 0.1	92.2 \pm 0.2	94.0 \pm 0.2	91.6 \pm 0.2
Cable	42.9 \pm 0.0	63.5 \pm 0.7	59.1 \pm 3.2	84.7 \pm 1.0	72.5 \pm 2.3	66.7 \pm 0.9	66.5 \pm 2.8	88.5 \pm 0.9	74.7 \pm 2.3	69.5 \pm 0.4	74.2 \pm 1.8	91.7 \pm 0.6	77.0 \pm 1.1
Capsule	62.1 \pm 0.0	92.7 \pm 0.4	80.0 \pm 2.0	83.9 \pm 0.9	85.6 \pm 2.7	93.4 \pm 0.3	82.3 \pm 2.1	86.6 \pm 1.0	90.6 \pm 0.6	94.1 \pm 0.6	85.7 \pm 1.3	87.8 \pm 1.9	90.1 \pm 1.5
Carpet	84.1 \pm 0.0	96.1 \pm 0.0	92.9 \pm 0.3	93.3 \pm 0.3	97.4 \pm 0.4	96.2 \pm 0.0	93.9 \pm 0.2	93.7 \pm 0.4	97.3 \pm 0.3	96.3 \pm 0.0	94.4 \pm 0.2	93.9 \pm 0.4	97.0 \pm 0.2
Grid	57.0 \pm 0.0	67.7 \pm 1.9	41.2 \pm 4.6	21.7 \pm 9.5	90.5 \pm 2.7	72.1 \pm 1.5	45.1 \pm 3.6	23.7 \pm 3.8	92.8 \pm 2.5	78.0 \pm 1.5	55.5 \pm 3.4	30.4 \pm 4.6	93.6 \pm 0.6
Hazelnut	81.6 \pm 0.0	94.9 \pm 0.3	85.7 \pm 1.9	88.3 \pm 1.3	93.7 \pm 0.9	95.6 \pm 0.2	89.4 \pm 0.9	89.8 \pm 1.3	94.2 \pm 0.3	95.6 \pm 0.1	90.4 \pm 0.7	92.0 \pm 0.3	94.2 \pm 0.3
Leather	91.1 \pm 0.0	98.7 \pm 0.0	95.6 \pm 0.2	95.2 \pm 1.0	98.6 \pm 0.0	98.8 \pm 0.0	96.2 \pm 0.2	95.9 \pm 0.3	98.3 \pm 0.4	98.8 \pm 0.0	96.3 \pm 0.1	96.4 \pm 0.1	98.0 \pm 0.4
Metal nut	31.8 \pm 0.0	73.4 \pm 1.1	38.1 \pm 1.6	66.7 \pm 2.9	84.7 \pm 1.1	78.1 \pm 1.8	48.2 \pm 5.0	79.6 \pm 4.2	86.7 \pm 0.8	81.2 \pm 1.4	54.0 \pm 8.8	83.8 \pm 5.5	89.4 \pm 0.1
Pill	65.0 \pm 0.0	92.8 \pm 0.3	78.9 \pm 0.6	89.5 \pm 1.6	93.5 \pm 0.2	93.3 \pm 0.2	84.3 \pm 0.4	91.6 \pm 0.5	94.5 \pm 0.2	93.9 \pm 0.2	86.6 \pm 0.4	92.5 \pm 0.4	94.6 \pm 0.3
Screw	68.5 \pm 0.0	85.0 \pm 0.8	51.6 \pm 1.7	68.1 \pm 1.3	82.3 \pm 1.1	87.2 \pm 1.2	69.5 \pm 2.1	69.0 \pm 2.1	84.1 \pm 0.5	89.5 \pm 1.3	72.3 \pm 0.8	72.4 \pm 3.1	86.3 \pm 1.8
Tile	51.2 \pm 0.0	84.2 \pm 0.4	66.7 \pm 1.5	82.5 \pm 1.1	89.4 \pm 0.4	84.6 \pm 0.2	71.9 \pm 0.5	82.5 \pm 0.5	89.6 \pm 0.4	84.9 \pm 0.1	73.6 \pm 0.9	83.0 \pm 0.1	89.9 \pm 0.3
Toothbrush	67.7 \pm 0.0	83.5 \pm 1.3	82.1 \pm 1.5	79.0 \pm 2.4	85.3 \pm 1.0	87.4 \pm 1.1	83.3 \pm 2.6	81.0 \pm 0.7	84.7 \pm 1.4	89.0 \pm 1.1	87.1 \pm 1.7	85.5 \pm 3.0	86.0 \pm 3.3
Transistor	43.4 \pm 0.0	55.3 \pm 2.0	70.3 \pm 7.0	70.9 \pm 4.6	65.0 \pm 1.8	57.6 \pm 1.4	76.5 \pm 5.5	78.8 \pm 1.5	68.6 \pm 1.1	58.5 \pm 0.7	82.2 \pm 7.4	79.5 \pm 2.8	69.0 \pm 1.1
Wood	74.1 \pm 0.0	92.9 \pm 0.1	86.5 \pm 0.6	87.1 \pm 1.0	91.0 \pm 0.6	93.1 \pm 0.1	88.0 \pm 0.2	86.8 \pm 1.4	91.8 \pm 0.6	93.2 \pm 0.1	88.4 \pm 0.2	87.7 \pm 0.4	91.7 \pm 0.3
Zipper	71.7 \pm 0.0	86.8 \pm 0.6	81.7 \pm 2.0	91.2 \pm 1.1	86.0 \pm 1.7	89.0 \pm 0.4	85.6 \pm 0.7	92.8 \pm 0.4	86.4 \pm 1.6	90.1 \pm 0.2	87.2 \pm 0.8	93.4 \pm 0.2	86.9 \pm 0.7
Mean	64.6\pm0.0	83.9 \pm 0.7	73.3 \pm 2.0	79.7 \pm 2.0	87.1\pm1.2	85.7 \pm 0.7	78.2 \pm 1.8	82.3 \pm 1.3	88.4\pm0.9	87.0 \pm 0.5	81.3 \pm 1.9	84.3 \pm 1.6	89.0\pm0.8

Table 14. Comparison of anomaly segmentation (AS) performance in terms of class-wise PRO on MVTec-AD. We report the mean and standard deviation over 5 random seeds for each measurement.

VisA (AC)	$K = 0$	$K = 1$				$K = 2$				$K = 4$			
		SPADE	PaDiM	PatchCore	WinCLIP+	SPADE	PaDiM	PatchCore	WinCLIP+	SPADE	PaDiM	PatchCore	WinCLIP+
Candle	95.4 \pm 0.0	86.1 \pm 5.6	70.8 \pm 4.1	85.1 \pm 1.4	93.4 \pm 1.4	91.3 \pm 3.3	75.8 \pm 2.1	85.3 \pm 1.5	94.8 \pm 1.0	92.8 \pm 2.1	77.5 \pm 1.6	87.8 \pm 0.8	95.1 \pm 0.3
Capsules	85.0 \pm 0.0	73.3 \pm 7.5	51.0 \pm 7.8	60.0 \pm 7.6	85.0 \pm 3.1	71.7 \pm 11.2	51.7 \pm 4.6	57.8 \pm 5.4	84.9 \pm 0.8	73.4 \pm 7.1	52.7 \pm 3.4	63.4 \pm 5.4	86.8 \pm 1.7
Cashew	92.1 \pm 0.0	95.9 \pm 1.1	62.3 \pm 9.9	89.5 \pm 4.4	94.0 \pm 0.4	97.3 \pm 1.4	74.6 \pm 3.6	93.6 \pm 0.6	94.3 \pm 0.5	96.4 \pm 1.3	77.7 \pm 3.2	93.0 \pm 1.5	95.2 \pm 0.8
Chewinggum	96.5 \pm 0.0	92.1 \pm 2.0	69.9 \pm 4.9	97.3 \pm 0.3	97.6 \pm 0.8	93.4 \pm 1.0	82.7 \pm 2.1	97.8 \pm 0.6	97.3 \pm 0.8	93.5 \pm 1.4	83.5 \pm 3.7	98.3 \pm 0.3	97.7 \pm 0.3
Fryum	80.3 \pm 0.0	81.1 \pm 4.0	58.3 \pm 5.9	75.0 \pm 4.8	88.5 \pm 1.9	90.5 \pm 3.9	69.2 \pm 9.0	83.4 \pm 2.4	90.5 \pm 0.4	92.9 \pm 1.6	71.2 \pm 5.9	88.6 \pm 1.3	90.8 \pm 0.5
Macaroni1	76.2 \pm 0.0	66.0 \pm 10.5	62.1 \pm 4.6	68.0 \pm 3.4	82.9 \pm 1.5	69.1 \pm 8.2	62.2 \pm 5.0	75.6 \pm 4.6	83.3 \pm 1.9	65.8 \pm 1.2	65.9 \pm 3.9	82.9 \pm 2.7	85.2 \pm 0.9
Macaroni2	63.7 \pm 0.0	55.8 \pm 6.1	47.5 \pm 5.9	55.6 \pm 4.6	70.2 \pm 0.9	58.3 \pm 4.4	50.8 \pm 2.9	57.3 \pm 5.6	71.8 \pm 2.0	56.7 \pm 3.2	55.0 \pm 2.9	61.7 \pm 1.8	70.9 \pm 2.2
PCB1	73.6 \pm 0.0	87.2 \pm 2.3	76.2 \pm 1.2	78.9 \pm 1.1	75.6 \pm 23.0	86.7 \pm 1.1	62.4 \pm 10.8	71.5 \pm 20.0	76.7 \pm 5.2	83.4 \pm 8.5	82.6 \pm 1.5	84.7 \pm 6.7	88.3 \pm 1.7
PCB2	51.2 \pm 0.0	73.5 \pm 3.7	61.2 \pm 2.0	81.5 \pm 0.8	62.2 \pm 3.9	70.3 \pm 8.1	66.8 \pm 2.0	84.3 \pm 1.7	62.6 \pm 3.7	71.7 \pm 7.0	73.5 \pm 2.4	84.3 \pm 1.0	67.5 \pm 2.6
PCB3	73.4 \pm 0.0	72.2 \pm 1.0	51.4 \pm 12.2	82.7 \pm 2.3	74.1 \pm 1.1	75.8 \pm 5.7	67.3 \pm 3.8	84.8 \pm 1.2	78.8 \pm 1.9	79.0 \pm 4.1	65.9 \pm 1.9	87.0 \pm 1.1	83.3 \pm 1.7
PCB4	79.6 \pm 0.0	93.4 \pm 1.3	76.1 \pm 3.6	93.9 \pm 2.8	85.2 \pm 8.9	86.1 \pm 8.2	69.3 \pm 13.7	94.3 \pm 3.2	82.3 \pm 9.9	95.4 \pm 2.3	85.4 \pm 2.0	95.6 \pm 1.6	87.6 \pm 8.0
Pipe fryum	69.7 \pm 0.0	77.9 \pm 3.2	66.7 \pm 2.2	90.7 \pm 1.7	97.2 \pm 1.1	78.1 \pm 3.0	75.3 \pm 1.8	93.5 \pm 1.3	98.0 \pm 0.6	79.3 \pm 0.9	82.9 \pm 2.2	96.4 \pm 0.7	98.5 \pm 0.4
Mean	78.1\pm0.0	79.5 \pm 4.0	62.8 \pm 5.4	79.9 \pm 2.9	83.8\pm4.0	80.7 \pm 5.0	67.4 \pm 5.1	81.6 \pm 4.0	84.6\pm2.4	81.7 \pm 3.4	72.8 \pm 2.9	85.3 \pm 2.1	87.3\pm1.8

Table 16. Comparison of anomaly classification (AC) performance in terms of class-wise AUROC on VisA. We report the mean and standard deviation over 5 random seeds for each measurement.

VisA (AC)	$K = 0$	$K = 1$				$K = 2$				$K = 4$			
		SPADE	PaDiM	PatchCore	WinCLIP+	SPADE	PaDiM	PatchCore	WinCLIP+	SPADE	PaDiM	PatchCore	WinCLIP+
Candle	95.8 \pm 0.0	86.5 \pm 4.3	69.2 \pm 3.9	86.6 \pm 2.3	93.6 \pm 1.5	90.7 \pm 3.2	72.8 \pm 1.0	86.8 \pm 1.7	95.1 \pm 1.1	92.6 \pm 1.9	72.5 \pm 1.1	88.9 \pm 1.1	95.3 \pm 0.4
Capsules	90.9 \pm 0.0	79.4 \pm 4.9	63.4 \pm 5.7	72.3 \pm 5.3	89.9 \pm 2.5	79.9 \pm 5.8	63.4 \pm 2.0	73.6 \pm 4.7	88.9 \pm 0.7	81.1 \pm 4.5	63.0 \pm 2.3	78.4 \pm 3.1	91.5 \pm 1.4
Cashew	96.4 \pm 0.0	97.9 \pm 0.4	78.2 \pm 5.7	94.6 \pm 2.0	97.2 \pm 0.2	98.6 \pm 0.6	86.1 \pm 2.2	96.9 \pm 0.3	97.3 \pm 0.2	98.3 \pm 0.6	88.4 \pm 2.0	96.5 \pm 0.7	97.7 \pm 0.4
Chewinggum	98.6 \pm 0.0	96.4 \pm 0.9	79.8 \pm 3.6	98.9 \pm 0.1	99.0 \pm 0.3	97.1 \pm 0.4	89.5 \pm 1.9	99.1 \pm 0.2	98.9 \pm 0.3	97.1 \pm 0.6	88.5 \pm 3.2	99.3 \pm 0.1	99.0 \pm 0.1
Fryum	90.1 \pm 0.0	88.9 \pm 1.8	74.5 \pm 2.9	87.6 \pm 2.4	94.7 \pm 1.0	94.5 \pm 2.3	81.0 \pm 5.4	92.1 \pm 1.3	95.8 \pm 0.2	95.8 \pm 1.0	81.5 \pm 3.0	95.0 \pm 0.6	96.0 \pm 0.3
Macaroni1	75.8 \pm 0.0	61.9 \pm 11.2	60.4 \pm 2.9	67.8 \pm 3.4	84.9 \pm 1.2	64.5 \pm 9.5	63.1 \pm 4.3	74.9 \pm 5.2	84.7 \pm 1.5	60.2 \pm 2.7	64.9 \pm 2.1	82.1 \pm 3.5	86.5 \pm 0.6
Macaroni2	60.3 \pm 0.0	52.7 \pm 4.2	51.7 \pm 5.0	54.9 \pm 3.2	68.4 \pm 1.8	55.9 \pm 3.1	52.7 \pm 1.5	57.2 \pm 2.6	70.4 \pm 1.8	51.9 \pm 2.3	54.9 \pm 2.5	60.2 \pm 3.0	69.6 \pm 2.8
PCB1	78.4 \pm 0.0	84.9 \pm 3.7	68.6 \pm 2.4	72.1 \pm 2.5	76.5 \pm 19.0	83.8 \pm 2.1	60.4 \pm 7.7	72.6 \pm 16.4	78.3 \pm 4.3	83.2 \pm 7.2	77.4 \pm 2.9	81.0 \pm 9.2	87.7 \pm 1.7
PCB2	49.2 \pm 0.0	74.9 \pm 2.9	63.3 \pm 1.2	84.4 \pm 0.4	64.9 \pm 3.3	71.7 \pm 6.6	68.9 \pm 2.6	86.6 \pm 1.1	65.8 \pm 4.0	74.2 \pm 5.0	75.0 \pm 1.7	86.2 \pm 1.0	71.3 \pm 3.4
PCB3	76.5 \pm 0.0	75.5 \pm 2.1	52.3 \pm 10.8	84.6 \pm 1.5	73.5 \pm 1.6	78.3 \pm 5.2	65.2 \pm 3.8	86.1 \pm 0.5	80.9 \pm 1.6	81.0 \pm 3.6	64.5 \pm 2.4	88.3 \pm 1.1	84.8 \pm 1.8
PCB4	77.7 \pm 0.0	92.9 \pm 1.6	74.7 \pm 2.6	92.8 \pm 3.1	78.5 \pm 15.5	81.9 \pm 11.2	67.6 \pm 11.9	93.2 \pm 3.4	72.5 \pm 16.2	94.8 \pm 2.9	84.0 \pm 2.0	94.9 \pm 1.2	85.6 \pm 8.9
Pipe fryum	82.3 \pm 0.0	88.3 \pm 2.0	79.2 \pm 1.5	95.4 \pm 0.6	98.6 \pm 0.5	88.1 \pm 1.7	84.5 \pm 1.7	96.8 \pm 0.7	99.0 \pm 0.3	88.8 \pm 1.0	89.8 \pm 1.7	98.3 \pm 0.3	99.2 \pm 0.2
Mean	81.2\pm0.0	82.0 \pm 3.3	68.3 \pm 4.0	82.8 \pm 2.3	85.1\pm4.0	82.3 \pm 4.3	71.6 \pm 3.8	84.8 \pm 3.2	85.8\pm2.7	83.4 \pm 2.7	75.6 \pm 2.2	87.5 \pm 2.1	88.8\pm1.8

Table 17. Comparison of anomaly classification (AC) performance in terms of class-wise AUPR on VisA. We report the mean and standard deviation over 5 random seeds for each measurement.

VisA (AC)	$K = 0$	$K = 1$				$K = 2$				$K = 4$			
		SPADE	PaDiM	PatchCore	WinCLIP+	SPADE	PaDiM	PatchCore	WinCLIP+	SPADE	PaDiM	PatchCore	WinCLIP+
Candle	89.4 \pm 0.0	80.4 \pm 6.1	72.0 \pm 2.0	79.5 \pm 0.6	87.8 \pm 1.2	85.5 \pm 2.9	74.8 \pm 2.9	78.7 \pm 0.8	89.1 \pm 1.3	86.9 \pm 2.4	76.7 \pm 1.3	80.5 \pm 0.9	88.9 \pm 1.0
Capsules	83.9 \pm 0.0	81.0 \pm 2.4	75.5 \pm 0.8	77.9 \pm 1.5	84.9 \pm 2.0	80.4 \pm 3.9	77.2 \pm 0.4	77.0 \pm 0.3	85.4 \pm 0.6	79.5 \pm 1.8	77.2 \pm 0.3	77.3 \pm 0.6	86.0 \pm 0.9
Cashew	88.4 \pm 0.0	94.8 \pm 1.8	80.8 \pm 0.7	89.6 \pm 3.6	90.7 \pm 0.7	95.5 \pm 2.2	82.4 \pm 1.2	92.3 \pm 0.5	90.9 \pm 0.7	95.7 \pm 0.9	82.5 \pm 1.2	91.1 \pm 2.1	91.6 \pm 1.3
Chewinggum	94.8 \pm 0.0	89.7 \pm 2.2	83.8 \pm 2.0	95.9 \pm 0.8	95.6 \pm 0.9	90.5 \pm 1.2	86.7 \pm 0.8	97.0 \pm 0.5	95.4 \pm 0.6	91.3 \pm 1.5	87.9 \pm 0.8	97.4 \pm 0.6	95.7 \pm 0.5
Fryum	82.7 \pm 0.0	85.3 \pm 2.1	80.6 \pm 0.8	82.9 \pm 1.7	87.2 \pm 1.4	90.9 \pm 1.8	82.7 \pm 2.1	84.7 \pm 1.4	88.4 \pm 0.6	91.9 \pm 1.7	82.9 \pm 1.8	86.6 \pm 0.6	88.9 \pm 0.8
Macaroni1	74.2 \pm 0.0	71.9 \pm 2.0	69.2 \pm 2.3	70.4 \pm 1.9	76.2 \pm 1.4	72.8 \pm 3.1	68.8 \pm 1.8	74.3 \pm 2.1	76.7 \pm 2.0	70.8 \pm 1.2	70.0 \pm 1.6	78.9 \pm 1.4	78.2 \pm 1.2
Macaroni2	69.8 \pm 0.0	68.1 \pm 0.8	67.1 \pm 0.2	67.6 \pm 0.7	72.3 \pm 1.1	68.2 \pm 1.2	67.1 \pm 0.4	67.6 \pm 1.2	73.9 \pm 0.9	67.9 \pm 0.6	68.4 \pm 1.0	68.8 \pm 0.8	73.1 \pm 1.6
PCB1	71.0 \pm 0.0	85.5 \pm 0.2	80.3 \pm 0.8	84.5 \pm 0.4	81.3 \pm 6.6	85.8 \pm 0.2	71.4 \pm 5.4	78.3 \pm 6.7	73.2 \pm 3.7	81.2 \pm 6.4	83.1 \pm 0.6	85.6 \pm 1.8	83.1 \pm 2.2
PCB2	67.1 \pm 0.0	70.9 \pm 1.9	68.6 \pm 1.4	75.9 \pm 0.8	67.2 \pm 0.3	71.5 \pm 2.7	69.2 \pm 0.6	78.1 \pm 2.1	67.3 \pm 0.3	71.1 \pm 3.2	72.0 \pm 2.3	79.2 \pm 1.9	67.7 \pm 0.6
PCB3	71.0 \pm 0.0	70.2 \pm 1.4											

VisA (AS)	$K = 0$	$K = 1$				$K = 2$				$K = 4$			
		SPADE	PaDiM	PatchCore	WinCLIP+	SPADE	PaDiM	PatchCore	WinCLIP+	SPADE	PaDiM	PatchCore	WinCLIP+
Candle	88.9 \pm 0.0	97.9 \pm 0.3	91.7 \pm 2.2	97.2 \pm 0.2	97.4 \pm 0.2	98.1 \pm 0.2	94.9 \pm 0.8	97.7 \pm 0.3	97.7 \pm 0.1	98.2 \pm 0.1	95.4 \pm 0.2	97.9 \pm 0.1	97.8 \pm 0.2
Capsules	81.6 \pm 0.0	95.5 \pm 0.5	70.9 \pm 1.1	93.2 \pm 0.9	96.4 \pm 0.6	96.5 \pm 0.9	75.7 \pm 1.7	94.0 \pm 0.2	96.8 \pm 0.3	97.7 \pm 0.1	79.1 \pm 0.7	94.8 \pm 0.5	97.1 \pm 0.2
Cashew	84.7 \pm 0.0	95.9 \pm 0.5	95.5 \pm 0.6	98.1 \pm 0.1	98.5 \pm 0.2	95.9 \pm 0.4	96.4 \pm 0.4	98.2 \pm 0.2	98.5 \pm 0.1	95.9 \pm 0.3	97.2 \pm 0.3	98.3 \pm 0.2	98.7 \pm 0.0
Chewinggum	93.3 \pm 0.0	96.0 \pm 0.4	90.1 \pm 0.4	96.9 \pm 0.3	98.6 \pm 0.1	96.0 \pm 0.3	93.1 \pm 0.7	96.6 \pm 0.1	98.6 \pm 0.1	95.7 \pm 0.3	94.4 \pm 0.5	96.8 \pm 0.1	98.5 \pm 0.1
Fryum	88.5 \pm 0.0	93.5 \pm 0.3	93.3 \pm 0.6	93.3 \pm 0.5	96.4 \pm 0.3	93.9 \pm 0.2	94.1 \pm 0.6	94.0 \pm 0.3	97.0 \pm 0.2	94.4 \pm 0.1	95.0 \pm 0.4	94.2 \pm 0.2	97.1 \pm 0.1
Macaroni1	70.9 \pm 0.0	97.9 \pm 0.2	89.4 \pm 0.9	95.2 \pm 0.4	96.4 \pm 0.6	98.5 \pm 0.2	91.7 \pm 0.3	96.0 \pm 1.3	96.5 \pm 0.7	98.8 \pm 0.1	93.5 \pm 0.5	97.0 \pm 0.3	97.0 \pm 0.2
Macaroni2	59.3 \pm 0.0	94.1 \pm 1.0	86.4 \pm 1.1	89.1 \pm 1.6	96.8 \pm 0.4	95.2 \pm 0.4	90.1 \pm 0.8	90.2 \pm 1.9	96.8 \pm 0.6	96.4 \pm 0.2	90.2 \pm 0.3	93.9 \pm 0.3	97.3 \pm 0.3
PCB1	61.2 \pm 0.0	94.7 \pm 0.4	89.9 \pm 0.3	96.1 \pm 1.5	96.6 \pm 0.6	96.5 \pm 1.5	90.6 \pm 0.6	97.6 \pm 0.9	97.0 \pm 0.9	96.8 \pm 1.5	93.2 \pm 1.5	98.1 \pm 1.0	98.1 \pm 0.9
PCB2	71.6 \pm 0.0	95.1 \pm 0.2	90.9 \pm 1.4	95.4 \pm 0.2	93.0 \pm 0.4	95.7 \pm 0.1	93.9 \pm 0.9	96.0 \pm 0.3	93.9 \pm 0.2	96.3 \pm 0.0	93.7 \pm 1.0	96.6 \pm 0.2	94.6 \pm 0.4
PCB3	85.3 \pm 0.0	96.0 \pm 0.1	93.9 \pm 0.3	96.2 \pm 0.3	94.3 \pm 0.3	96.6 \pm 0.1	95.1 \pm 0.5	97.1 \pm 0.1	95.1 \pm 0.2	96.9 \pm 0.0	95.7 \pm 0.1	97.4 \pm 0.2	95.8 \pm 0.1
PCB4	94.4 \pm 0.0	92.0 \pm 0.6	89.6 \pm 0.6	95.6 \pm 0.6	94.0 \pm 0.9	92.8 \pm 0.3	90.7 \pm 0.9	96.2 \pm 0.4	95.6 \pm 0.3	94.1 \pm 0.2	92.1 \pm 0.5	97.0 \pm 0.2	96.1 \pm 0.3
Pipe fryum	75.4 \pm 0.0	98.4 \pm 0.2	97.2 \pm 2.6	98.8 \pm 0.2	98.3 \pm 0.2	98.7 \pm 0.1	98.1 \pm 0.4	99.1 \pm 0.1	98.5 \pm 0.2	98.8 \pm 0.0	98.5 \pm 0.1	99.1 \pm 0.0	98.7 \pm 0.1
Mean	79.6\pm0.0	95.6 \pm 0.4	89.9 \pm 0.8	95.4 \pm 0.6	96.4\pm0.4	96.2 \pm 0.4	92.0 \pm 0.7	96.1 \pm 0.5	96.8\pm0.3	96.6 \pm 0.3	93.2 \pm 0.5	96.8 \pm 0.3	97.2\pm0.2

Table 19. Comparison of anomaly segmentation (AS) performance in terms of class-wise pixel-AUROC on VisA. We report the mean and standard deviation over 5 random seeds for each measurement.

VisA (AS)	$K = 0$	$K = 1$				$K = 2$				$K = 4$			
		PRO	WinCLIP	SPADE	PaDiM	PatchCore	WinCLIP+	SPADE	PaDiM	PatchCore	WinCLIP+	SPADE	PaDiM
Candle	83.5 \pm 0.0	95.6 \pm 0.5	81.5 \pm 5.3	92.6 \pm 0.4	94.0 \pm 0.4	95.6 \pm 0.4	87.3 \pm 1.2	93.4 \pm 0.6	94.2 \pm 0.2	95.7 \pm 0.1	88.3 \pm 0.7	94.1 \pm 0.4	94.4 \pm 0.2
Capsules	35.3 \pm 0.0	83.1 \pm 1.1	30.6 \pm 1.1	66.6 \pm 4.5	73.6 \pm 3.5	85.4 \pm 3.1	38.4 \pm 3.7	67.9 \pm 2.3	75.9 \pm 1.9	89.0 \pm 1.2	43.3 \pm 2.0	69.0 \pm 3.2	77.0 \pm 1.4
Cashew	76.4 \pm 0.0	89.8 \pm 1.1	73.4 \pm 2.1	90.8 \pm 0.2	91.1 \pm 0.8	90.4 \pm 0.5	78.4 \pm 2.7	91.4 \pm 1.0	90.4 \pm 0.6	90.4 \pm 0.6	81.2 \pm 2.8	92.1 \pm 0.3	91.3 \pm 0.9
Chewinggum	70.4 \pm 0.0	73.9 \pm 1.2	58.1 \pm 0.6	78.2 \pm 1.3	91.0 \pm 0.5	73.8 \pm 1.1	63.7 \pm 2.4	78.0 \pm 0.4	90.9 \pm 0.7	72.7 \pm 0.9	67.2 \pm 1.8	79.3 \pm 0.8	91.0 \pm 0.4
Fryum	77.4 \pm 0.0	83.7 \pm 1.2	71.1 \pm 1.6	78.7 \pm 2.3	89.1 \pm 1.0	84.5 \pm 0.9	71.2 \pm 0.8	81.4 \pm 2.8	89.3 \pm 0.2	86.2 \pm 0.9	73.2 \pm 1.3	81.0 \pm 1.2	89.7 \pm 0.5
Macaroni1	34.3 \pm 0.0	92.0 \pm 0.6	62.2 \pm 4.4	83.4 \pm 1.3	84.6 \pm 2.3	93.9 \pm 0.8	71.8 \pm 2.4	86.2 \pm 4.6	85.2 \pm 1.4	95.1 \pm 0.4	76.6 \pm 2.1	89.6 \pm 0.7	86.8 \pm 0.8
Macaroni2	21.4 \pm 0.0	80.0 \pm 3.3	54.9 \pm 3.6	66.0 \pm 3.0	89.3 \pm 2.4	81.7 \pm 1.5	65.6 \pm 3.4	67.2 \pm 6.5	88.6 \pm 1.7	86.0 \pm 0.8	65.9 \pm 1.5	78.3 \pm 0.9	90.5 \pm 1.3
PCB1	26.3 \pm 0.0	81.3 \pm 5.7	63.9 \pm 1.8	79.0 \pm 10.7	82.5 \pm 6.0	87.2 \pm 2.3	68.4 \pm 4.1	86.1 \pm 1.7	83.8 \pm 5.0	88.0 \pm 2.7	70.2 \pm 3.3	88.1 \pm 2.6	87.9 \pm 2.1
PCB2	37.2 \pm 0.0	83.7 \pm 0.6	64.4 \pm 3.8	80.9 \pm 0.5	73.6 \pm 1.5	85.5 \pm 1.0	72.9 \pm 3.4	82.9 \pm 1.8	76.2 \pm 0.9	87.0 \pm 0.5	71.9 \pm 2.6	83.7 \pm 1.0	78.0 \pm 1.3
PCB3	56.1 \pm 0.0	84.3 \pm 1.0	69.0 \pm 1.2	78.1 \pm 2.0	79.5 \pm 2.5	86.1 \pm 0.6	74.0 \pm 2.3	82.2 \pm 1.1	82.3 \pm 1.8	87.7 \pm 0.6	77.2 \pm 0.8	84.4 \pm 1.9	84.2 \pm 1.0
PCB4	80.4 \pm 0.0	66.9 \pm 2.0	59.1 \pm 1.8	77.9 \pm 3.1	76.6 \pm 4.1	69.3 \pm 1.1	62.6 \pm 3.6	79.5 \pm 4.8	81.7 \pm 1.2	74.7 \pm 1.0	67.9 \pm 2.6	83.5 \pm 2.5	84.2 \pm 0.7
Pipe fryum	82.3 \pm 0.0	94.3 \pm 0.5	83.9 \pm 0.8	93.6 \pm 0.5	96.1 \pm 0.6	95.0 \pm 0.2	86.9 \pm 0.9	94.5 \pm 0.4	96.2 \pm 0.6	95.0 \pm 0.3	88.7 \pm 1.3	95.0 \pm 0.5	96.6 \pm 0.2
Mean	56.8\pm0.0	84.1 \pm 1.6	64.3 \pm 2.4	80.5 \pm 2.5	85.1\pm2.1	85.7 \pm 1.1	70.1 \pm 2.6	82.6 \pm 2.3	86.2\pm1.4	87.3 \pm 0.8	72.6 \pm 1.9	84.9 \pm 1.4	87.6\pm0.9

Table 20. Comparison of anomaly segmentation (AS) performance in terms of class-wise PRO on VisA. We report the mean and standard deviation over 5 random seeds for each measurement.

VisA (AS)	$K = 0$	$K = 1$				$K = 2$				$K = 4$			
		F_1 -max	WinCLIP	SPADE	PaDiM	PatchCore	WinCLIP+	SPADE	PaDiM	PatchCore	WinCLIP+	SPADE	PaDiM
Candle	22.5 \pm 0.0	37.1 \pm 0.8	19.6 \pm 2.6	40.9 \pm 1.0	42.7 \pm 1.7	37.6 \pm 0.5	21.6 \pm 1.4	40.4 \pm 1.0	42.2 \pm 0.8	38.3 \pm 1.0	21.6 \pm 0.7	41.0 \pm 1.0	43.0 \pm 0.9
Capsules	9.2 \pm 0.0	28.5 \pm 7.8	2.3 \pm 0.2	35.7 \pm 6.8	58.2 \pm 1.3	37.4 \pm 8.4	3.0 \pm 0.4	37.8 \pm 5.7	57.0 \pm 3.7	48.0 \pm 2.0	3.9 \pm 0.4	47.0 \pm 3.0	59.8 \pm 1.8
Cashew	13.2 \pm 0.0	51.1 \pm 1.4	35.1 \pm 4.9	60.4 \pm 1.1	59.5 \pm 2.1	52.8 \pm 1.2	40.8 \pm 4.0	60.3 \pm 0.4	60.5 \pm 2.4	54.1 \pm 0.5	47.2 \pm 2.9	60.7 \pm 0.4	62.3 \pm 1.1
Chewinggum	41.1 \pm 0.0	58.7 \pm 1.1	19.7 \pm 2.0	64.5 \pm 1.0	65.3 \pm 0.5	59.9 \pm 0.5	29.5 \pm 6.4	63.9 \pm 0.4	64.8 \pm 0.9	59.5 \pm 0.6	37.8 \pm 4.4	64.4 \pm 0.6	65.2 \pm 0.2
Fryum	22.1 \pm 0.0	34.0 \pm 1.5	32.3 \pm 1.5	37.2 \pm 1.4	50.8 \pm 1.8	36.6 \pm 2.1	36.5 \pm 3.6	41.1 \pm 2.9	54.8 \pm 1.7	40.3 \pm 1.8	44.5 \pm 1.3	44.6 \pm 2.9	56.5 \pm 0.6
Macaroni1	7.0 \pm 0.0	28.5 \pm 3.4	4.8 \pm 0.7	16.5 \pm 2.6	34.1 \pm 1.7	39.2 \pm 3.5	5.5 \pm 1.2	20.0 \pm 8.2	33.2 \pm 1.9	37.6 \pm 5.8	7.0 \pm 0.7	21.6 \pm 1.9	33.8 \pm 0.9
Macaroni2	1.0 \pm 0.0	6.7 \pm 2.8	1.9 \pm 0.5	2.7 \pm 0.7	34.4 \pm 3.0	8.6 \pm 1.8	2.4 \pm 0.1	5.1 \pm 2.3	29.9 \pm 3.4	18.3 \pm 3.0	2.4 \pm 0.3	10.9 \pm 1.7	35.1 \pm 2.5
PCB1	2.4 \pm 0.0	16.6 \pm 1.1	8.6 \pm 0.7	30.7 \pm 3.2	25.9 \pm 2.6	34.8 \pm 19.9	8.9 \pm 0.5	47.7 \pm 19.7	34.6 \pm 16.2	37.1 \pm 20.8	13.9 \pm 3.6	55.9 \pm 20.3	50.9 \pm 20.4
PCB2	4.7 \pm 0.0	35.0 \pm 1.9	9.8 \pm 2.2	33.3 \pm 1.0	18.7 \pm 1.5	39.2 \pm 1.1	16.6 \pm 2.5	33.3 \pm 0.6	24.0 \pm 1.1	42.6 \pm 1.0	15.8 \pm 3.0	33.6 \pm 0.3	27.8 \pm 1.9
PCB3	10.3 \pm												