

# FADE: Few-shot/zero-shot Anomaly Detection Engine using Large Vision-Language Model

Yuanwei Li<sup>1,2</sup>  
yuanwei\_li@hotmail.com

Elizaveta Ivanova<sup>1</sup>  
lisa.ivanova@entrust.com

Martins Bruveris  
martins.bruveris@entrust.com

Onfido  
London, UK

---

## Abstract

Automatic image anomaly detection is important for quality inspection in the manufacturing industry. The usual unsupervised anomaly detection approach is to train a model for each object class using a dataset of normal samples. However, a more realistic problem is zero-/few-shot anomaly detection where zero or only a few normal samples are available. This makes the training of object-specific models challenging. Recently, large foundation vision-language models have shown strong zero-shot performance in various downstream tasks. While these models have learned complex relationships between vision and language, they are not specifically designed for the tasks of anomaly detection. In this paper, we propose the Few-shot/zero-shot Anomaly Detection Engine (FADE) which leverages the vision-language CLIP model and adjusts it for the purpose of industrial anomaly detection. Specifically, we improve language-guided anomaly segmentation 1) by adapting CLIP to extract multi-scale image patch embeddings that are better aligned with language and 2) by automatically generating an ensemble of text prompts related to industrial anomaly detection. 3) We use additional vision-based guidance from the query and reference images to further improve both zero-shot and few-shot anomaly detection. On the MVTec-AD (and VisA) dataset, FADE outperforms other state-of-the-art methods in anomaly segmentation with pixel-AUROC of 89.6% (91.5%) in zero-shot and 95.4% (97.5%) in 1-normal-shot. Code is available at <https://github.com/BMVC-FADE/BMVC-FADE>.

## 1 Introduction

Industrial image anomaly detection involves anomaly classification (AC) and anomaly segmentation (AS) which aim to identify and localise anomalies occurring on objects found in manufacturing industries. Most of the current research focuses on unsupervised anomaly detection where only normal samples are used during training while both normal and anomalous samples are evaluated during inference [1, 2, 3, 4, 5, 6]. Some of these approaches

---

<sup>1</sup>These authors contributed equally.

<sup>2</sup>Corresponding author.

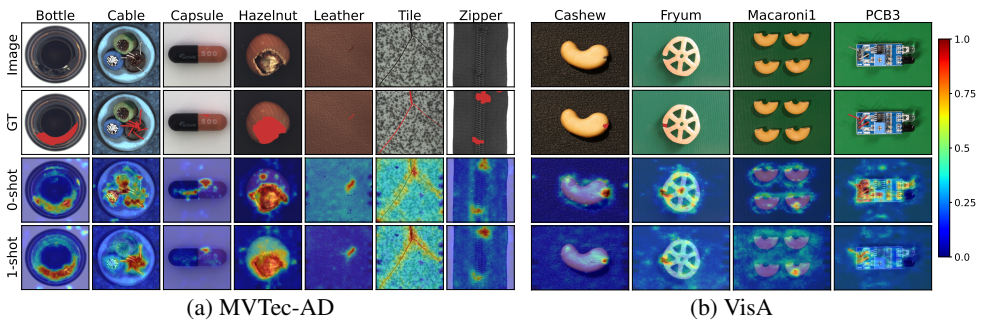


Figure 1: Zero-/one-shot anomaly segmentation results of FADE.

are able to achieve high performance when a large number of normal training samples is available. But this is not a realistic setting for industrial anomaly detection for two reasons. First, the number of normal training samples can be scarce or unavailable. Second, one specific model needs to be trained for each object class and this quickly becomes unscalable when the number of object classes increases.

Zero-shot/few-normal-shot anomaly detection is a more realistic setting with either zero or only a few normal reference images available. However, many previous methods designed for unsupervised anomaly detection perform poorly under this low-data regime. There are increasingly more research works focusing on this area recently [4, 9, 10, 22, 24] but there is still much room for improvement.

Vision foundation models trained on large image datasets have recently shown superb zero-shot capability on various computer vision tasks such as image classification, object detection and segmentation [11, 16, 17]. There is great potential to leverage and transfer the knowledge and representations learned in these foundation models for the zero-/few-shot anomaly detection task. But these foundation models are not specifically designed and trained for anomaly detection. As such, we propose the Few-shot/zero-shot Anomaly Detection Engine (FADE) that utilises one of these foundation models, the CLIP model [17], and adapt it for the purpose of industrial AC and AS.

CLIP is a visual-language model pretrained on large datasets of image-text pairs using a contrastive loss. CLIP learns concepts and representations that capture the relationship between language and image. Without any finetuning, the model has demonstrated zero-shot capability in various downstream tasks such as image classification. This is done by constructing free-form text prompts as classification labels which enable the use of natural language to extract relevant information already learned by the CLIP model. This allows CLIP to be a suitable candidate for handling the anomaly detection task. The concept of anomaly and normality is broad and abstract. But language allows us to describe a specific anomaly precisely (E.g. a cracked object, a scratched surface, a missing component). CLIP can leverage the power of language to better capture the concept of anomaly and hence improve upon zero-shot anomaly detection performance. However, this requires careful prompt engineering and some methods [10] manually design an ensemble of text prompts with key words such as “damage” and “defect” which can be a time-consuming process. As such, we propose to utilise existing Large Language Model (LLM) to automatically generate text prompts related to the concept of normality and anomaly.

There is another challenge associated with language-guide anomaly detection. ViT-based

CLIP model is trained by aligning the image-level CLS token embeddings with the text embeddings. This works for image-level anomaly classification but not pixel-level anomaly segmentation. This is because language-guided segmentation requires the comparison between the image patch embeddings and the text embeddings. However, the patch embeddings are not aligned to natural language during training. This results in sub-optimal segmentation performance [4, 10, 13]. To address this issue, we propose to apply the Grounding Everything Module (GEM) [9] to extract the image patch embeddings which are shown to have better alignment with language and perform better in zero-shot segmentation.

While CLIP can be used for language-driven anomaly detection, we can also only use its image encoder for vision-based anomaly detection. In this case, visual representations of image patches extracted from both query and reference images are compared to one another to identify inconsistencies and anomalies. This enhances zero-shot performance and extends the approach to the few-shot setting when normal reference images are available.

In summary, we propose FADE which uses pretrained CLIP model for zero-/few-shot anomaly detection without any further training or fine-tuning. Our main contributions are:

- We utilise GEM patch embeddings which are better aligned with language to improve zero-shot language-guided AS. In addition, we adopt a multi-scale approach to make the method robust at detecting anomalies of different sizes.
- We further improve language-guided anomaly detection by using an LLM to automatically generate a prompt ensemble that captures the concept of normality and anomaly using a diverse set of text prompts which are related to industrial anomaly inspection.
- We enhance anomaly detection performance by employing a vision-guided approach that can be applied to both zero- and few-shot settings.
- FADE shows competitive results on the MVTec-AD and VisA benchmarks for AC and AS under both the zero-shot and few-normal-shot regimes.

## 2 Related Work

**Anomaly detection:** Most research has focused on unsupervised anomaly detection where a model is trained on many normal training samples [2, 6, 8, 10, 13, 19]. Among them, PatchCore [13], a memory bank based method, has achieved state-of-the-art performance. There has been a growing interest in zero-/few-shot anomaly detection since the performance of the above methods decreases under such regimes. Earlier work such as Metaformer [22] addresses the problem in a meta-learning framework. RegAD [9] detects anomalies by comparing registered features of a test image with the normal reference images. However, both methods require model training using some datasets. A recent work, Segment Any Anomaly (SAA) [4], leverages a foundation segmentation model SAM [10] for zero-shot AS.

**Anomaly detection with CLIP:** Recently, CLIP has shown impressive language-driven zero-shot capability in various computer vision tasks [10]. WinCLIP [10] is one of the first to use CLIP for language-guided zero-/few-shot anomaly detection. The paper adapts CLIP for pixel-wise AS by extracting dense CLIP features based on overlapping windows. Subsequent papers APRIL-GAN [6] and AnomalyCLIP [22] build upon the general framework of WinCLIP and introduce additional learnable layers and learnable prompts respectively to fine-tune the model for AS. ClipSAM [13] leverages CLIP to obtain a rough anomaly segmentation mask from which points and bounding boxes are then extracted and used by SAM

as input prompts to further refine the segmentation. However, all the above methods use auxiliary datasets with ground truth anomaly segmentation masks for training. Other methods explore prompting in detail. [14] constructs manual and learnable normal and anomaly prompts and proposes a one-class prompt-learning method for few-shot detection. [15] augments normal and anomalous text prompts by inserting random words and uses their CLIP text embeddings to train a feed-forward network for zero-shot AC. Our method, unlike most of the above works, does not require further training or fine-tuning.

**Zero-shot segmentation with CLIP:** While CLIP shows strong zero-shot image classification capability, there are challenges to adapt it for dense pixel-wise segmentation. The main problem is that CLIP training does not directly optimise for alignment between language and local image patch embeddings [16]. In addition, empirical segmentation results show that CLIP tends to generate opposite visualisation between foreground/background and also produce noisy activations [15]. Several works have attempted to tackle these issues by adapting the CLIP model architecture for zero-shot segmentation [8, 15, 13]. CLIP Surgery [15] replaces the query-key attention in the original transformer block with a value-value attention and also removes the feed-forward network. [8] proposes GEM blocks which further generalises the idea to self-self attention. In our work, we apply the GEM blocks to the task of anomaly segmentation.

## 3 Method

FADE conducts both anomaly classification (AC) and anomaly segmentation (AS) under the zero-/few-shot settings. It consists of four different anomaly detection pipelines that are used under different cases (Fig. 2a-d). The detection pipeline can be either language- or vision-guided. Language-guided detection utilises the language capability of the CLIP model to perform zero-shot AC (Fig. 2a) and AS (Fig. 2b). On the other hand, vision-guided detection employs visual cues from images to further improve zero-shot AS (Fig. 2c) and extends the approach to few-shot detection (Fig. 2d).

### 3.1 Language-Guided Anomaly Classification

Zero-shot language-guided AC (Fig. 2a) is carried out as described in WinCLIP [10]. A prompt ensemble  $t = \{t_+, t_-\}$  is manually crafted which captures the notion of object anomaly or normality (Appendix A of [10]) where  $t_+$  and  $t_-$  are the sets of 88 anomalous and 154 normal text prompts respectively. An example of a normal prompt is “a cropped photo of the flawless [o]” while an anomalous prompt can be “a cropped photo of the damaged [o]” where [o] is the object label that comes with the dataset (E.g. bottle). The CLIP text encoder  $g$  extracts the text embeddings of all the prompts. The average text embeddings  $h$  for the anomalous and normal text prompts are computed as  $h_+ = \frac{1}{N_{t_+}} \sum_{t \in t_+} g(t)$  and  $h_- = \frac{1}{N_{t_-}} \sum_{t \in t_-} g(t)$  where  $N_{t_+}$  and  $N_{t_-}$  are the number of anomalous and normal prompts respectively. Given an image  $x$  and the CLIP image encoder  $f^{clip}$ , we compute its image embeddings  $f_{cls}^{clip}(x)$  where  $cls$  indicates the CLS token of the ViT-based CLIP model. The cosine similarity  $\langle f_{cls}^{clip}(x), h \rangle$  then gives a measure of how close the image is to the concept of normality or anomaly as encapsulated by the text prompts. Binary

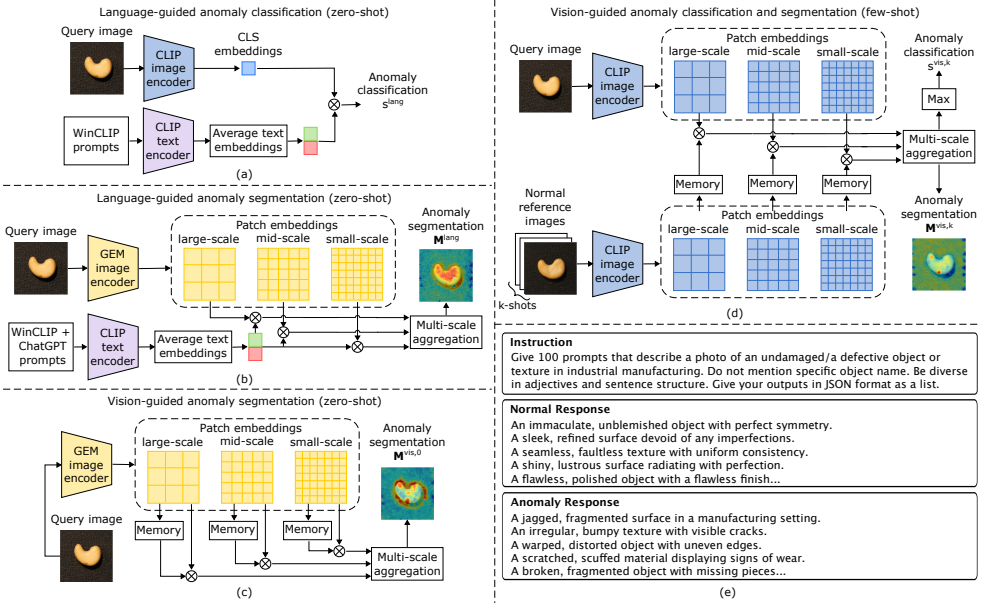


Figure 2: Different components of FADE. (a) Zero-shot language-guided AC; (b) Zero-shot language-guided AS; (c) Zero-shot vision-guided AS; (d) Few-shot vision-guided AC and AS; (e) ChatGPT prompts generation: An instruction given to ChatGPT and some of its responses.

zero-shot AC is performed by calculating a language-guided anomaly score:

$$s^{lang} = \frac{\exp(\langle f_{cls}^{clip}(x), h_+ \rangle / \tau)}{\exp(\langle f_{cls}^{clip}(x), h_+ \rangle / \tau) + \exp(\langle f_{cls}^{clip}(x), h_- \rangle / \tau)} \quad (1)$$

where  $\tau$  is a temperature parameter fixed at 0.01.

## 3.2 Language-Guided Anomaly Segmentation

**GEM embeddings:** Given an image of size  $h \times w$ , AS generates a dense pixel-level prediction  $M \in [0, 1]^{h \times w}$  which localises the anomalous regions. The above framework for AC can be extended to zero-shot language-guided AS (Fig. 2b). Specifically, the patch embeddings  $\{f_p^{clip}(x)\}_{p \in Patches}$  from the last transformer block of the CLIP image encoder are extracted to replace the CLS embeddings  $f_{cls}^{clip}(x)$  in Eq. 1, where *Patches* is a set of all patches extracted from an image  $x$ . This enables anomaly prediction at each location of a patch  $p$  that can be spatially combined to form a dense segmentation map. However, this approach yields poor results [8, 10] since CLIP is trained using an image-level contrastive loss that aligns the image-level CLS embeddings with the text embeddings. Hence, CLIP has poor alignment between its patch embeddings and the text embeddings, resulting in its failure to generalise its zero-shot capability to the dense segmentation task. WinCLIP [10] solves this issue by introducing a grid of overlapping windows where each window masks out the image content outside of the window and computes the CLS embeddings of the content inside the window. This produces a dense segmentation while maintaining the visual-language alignment.

We take a different approach based on the Grounding Everything Module (GEM) [9] that has demonstrated success in zero-shot open-vocabulary object localisation. GEM uses the idea of self-self attention which computes query-query, key-key and value-value attention weights instead of the query-key attention weights in the conventional self-attention mechanism. Mathematically, self-self attention weights are given by:  $Attn_{self-self} = \text{softmax}(h_{Patches}^{clip} W \cdot (h_{Patches}^{clip} W)^T)$  where  $h_{Patches}^{clip} \in \mathbb{R}^{N \times d}$  are the patch embeddings of dimension  $d$  from the previous transformer block of the original CLIP model for all  $N$  patches in an image.  $W \in \{W_v, W_q, W_k\}$  are the projection matrices of the current transformer block. The outputs from q-q, k-k and v-v attention are ensembled together to form the output of a GEM block. The GEM block also removes the feed-forward network from the conventional transformer block and is constructed as a parallel pathway to the original transformer blocks pathway. The outputs of all the GEM blocks at different layers are summed together to obtain the final GEM patch embeddings  $\{f_p^{gem}(x)\}_{p \in Patches}$ . The zero-shot language-guided AS mask  $M^{lang} = \{M_p^{lang}\}_{p \in Patches}$  is then obtained where each patch anomaly score  $M_p^{lang}$  is computed as:

$$M_p^{lang} = \frac{\exp(\langle f_p^{gem}(x), h_+ \rangle / \tau)}{\exp(\langle f_p^{gem}(x), h_+ \rangle / \tau) + \exp(\langle f_p^{gem}(x), h_- \rangle / \tau)} \quad (2)$$

GEM embeddings have shown better visual-language alignment without further fine-tuning the CLIP model. It also has a connection to clustering properties that group similar pixels together, resulting in better zero-shot segmentation [9]. GEM patch embeddings are computed based on the global context of the entire image which is important for anomaly segmentation (E.g. anomalous regions are better identified through comparison with normal regions). In contrast, WinCLIP embeddings only have a limited context restricted to a local window. In addition, GEM can be computationally more efficient with a single forward pass of the whole image.

**Prompt engineering:** CLIP is pretrained on large datasets of visual-language pairs and learns powerful representations and alignment between images and texts that are useful for anomaly detection. Prompt engineering that captures the specific description and concept of anomaly is essential for maximising the zero-shot capability of the CLIP model. While the WinCLIP prompt ensemble has been designed manually to optimise for anomaly detection, we notice that AS performance can be further improved with additional prompt engineering.

The concept of anomaly and normality is broad and goes beyond the limited descriptions provided by the WinCLIP prompts that use mostly general words like “damage”, “flaw”. In contrast, the MVTEC and VisA datasets contain different types of specific industrial anomalies such as “a scratch”, “a crack”, “a missing part”. While these specific descriptions are useful for AS, manually crafting them is time-consuming and unscalable. Instead, we use a large language model, ChatGPT 3.5<sup>1</sup>, to automatically generate a large number of text prompts that contain a diverse set of specific descriptions related to industrial anomaly detection. Fig. 2e shows an instruction prompt that is passed to ChatGPT 3.5 and some of its responses. The instruction ensures that specific object names are not mentioned in the response so that the generated text prompts are object-agnostic. This is because the notion of the specific object becomes less relevant for AS when the patch embeddings focus on local anomalous regions that are only part of the object. We also attempt to add information regarding the anomaly size and location in the text prompts but it did not improve the results.

<sup>1</sup><https://chat.openai.com>



We use 5 instruction prompts with different wordings to obtain responses that are more diverse and varied. This generates a total of 486 anomaly and 423 normal prompts that form the ChatGPT prompt ensemble. The instructions and their responses are listed in supplementary. The ChatGPT and WinCLIP prompt ensembles are combined and used for AS.

**Multi-scale aggregation:** We extract multi-scale GEM embeddings in order to detect anomalies with different sizes. Although the CLIP model is trained at a fixed image size, it can take inputs of different image sizes during inference by interpolating the positional embeddings. We resize the original image to 3 input sizes:  $240 \times 240$ ,  $448 \times 448$ ,  $896 \times 896$ . Given that the patch size of the model is 16, this corresponds to output segmentations with size  $15 \times 15$ ,  $28 \times 28$  and  $56 \times 56$ . The segmentations are then upscaled to a fixed size and averaged to give the final segmentation map  $M^{lang}$ .

### 3.3 Vision-Guided Anomaly Classification and Segmentation

Vision-guided anomaly detection extracts and compares visual cues from images in order to identify anomalies. This is complementary to the language-guided approach and further extends it to the few-shot setting.

**Few-shot setting:** The performance of AC and AS improves when a few reference normal images are available. Specifically, we follow the approach similar to WinCLIP [14] where we build a memory bank  $R$  of CLIP patch embeddings extracted from the  $k$  reference images (Fig. 2d). We have also tried to use GEM embeddings but the results are inferior. Given a query image  $x$  and its CLIP embeddings  $f_p^{clip}(x)$  for a patch  $p$ , the vision-guided anomaly score for this patch under the  $k$ -shot setting is computed as the cosine distance to its nearest neighbour patch in the memory bank:  $M_p^{vis,k} = \min_{r \in R} \frac{1}{2} \left( 1 - \langle f_p^{clip}(x), r \rangle \right)$ . Spatially combining the anomaly scores of the patches at all locations in an image gives the vision-guided segmentation map. We can extend this to a multi-scale approach by building a separate memory bank for the patch embeddings extracted from the reference images at different scales (E.g. different image sizes). The multi-scale segmentation maps are averaged to give the final vision-guided segmentation  $M^{vis,k}$  under the  $k$ -shot setting. In addition, we use the maximum value of  $M^{vis,k}$  as an anomaly score  $s^{vis,k}$  for few-shot vision-guided AC.

**Zero-shot setting:** Vision-guided AS can also be applied in the zero-shot setting (Fig. 2c) with 2 differences compared to the few-shot setting: 1) The memory bank is built using patches from the query image itself since reference images are unavailable. In this case, the resulting anomaly score is computed as the distance to the second nearest neighbour patch since the nearest neighbour is the patch itself. 2) GEM patch embeddings are used since they perform better than CLIP patch embeddings. Multi-scale aggregation is used again to give the final segmentation  $M^{vis,0}$  under the zero-shot setting.

**Combining language and vision guidance:** Language and vision guidance can be combined to improve overall AC and AS performance. Tab. 2 summarises how the combination is done under the different settings. Refer to supplementary for details.

## 4 Experiments

We conduct a series of experiments to evaluate the performance of FADE on AC and AS under the zero-/few-shot regime. We also perform ablation experiments to study the impact of each component in FADE.

Anomaly Classification		MVTec-AD			VisA		
Setup	Method	AUROC	AUPR	F1-max	AUROC	AUPR	F1-max
0-shot	WinCLIP	<b>91.8±0.0</b>	<b>96.5±0.0</b>	<b>92.9±0.0</b>	<b>78.1±0.0</b>	<b>81.2±0.0</b>	<b>79.0±0.0</b>
	FADE (ours)	90.0±0.0	95.6±0.0	92.4±0.0	75.6±0.0	78.5±0.0	78.6±0.0
1-shot	PatchCore	83.4±3.0	92.2±1.5	90.5±1.5	79.9±2.9	82.8±2.3	81.7±1.6
	WinCLIP+	93.1±2.0	96.5±0.9	93.7±1.1	83.8±4.0	85.1±4.0	83.1±1.7
	FADE (ours)	<b>93.9±0.7</b>	<b>96.8±0.3</b>	<b>94.8±0.2</b>	<b>86.7±2.0</b>	<b>87.9±1.5</b>	<b>84.7±0.8</b>
2-shot	PatchCore	86.3±3.3	93.8±1.7	92.0±1.5	81.6±4.0	84.8±3.2	82.5±1.8
	WinCLIP+	94.4±1.3	97.0±0.7	94.4±0.8	84.6±2.4	85.8±2.7	83.0±1.4
	FADE (ours)	<b>95.2±1.0</b>	<b>97.6±0.5</b>	<b>95.0±0.4</b>	<b>89.2±0.4</b>	<b>90.2±0.2</b>	<b>85.9±0.6</b>
4-shot	PatchCore	88.8±2.6	94.5±1.5	92.6±1.6	85.3±2.1	87.5±2.1	84.3±1.3
	WinCLIP+	95.2±1.3	97.3±0.6	94.7±0.8	87.3±1.8	88.8±1.8	84.2±1.6
	FADE (ours)	<b>96.3±0.4</b>	<b>98.1±0.2</b>	<b>95.5±0.4</b>	<b>90.7±0.3</b>	<b>91.9±0.4</b>	<b>87.0±0.2</b>

Table 1: Comparison of AC performance on MVTec-AD and VisA. We report the mean and standard deviation over 5 random seeds. Bold indicates the best performance.

Task	Shot	Aggregation
AC	0-shot	$s^{lang}$
	k-shot	$s^{lang} + s^{vis,k}$
AS	0-shot	$M^{lang} + M^{vis,0}$
	k-shot	$M^{lang} + M^{vis,k}$

Table 2: Language and vision aggregation.

Patch embeddings	MVTec-AD	VisA
CLIP	18.0	13.9
GEM	<b>86.5</b>	<b>87.0</b>

Table 3: CLIP vs GEM embeddings for 0-shot AS (pAUROC).

**Datasets:** All experiments are based on the MVTec-AD [11] and VisA [25] datasets which are standard benchmarks for AC and AS. The datasets include a range of different objects such as capsule and cashew. We only use the test split for evaluation which contains both normal and anomalous images and their ground truth segmentation masks. The training split is only used for sampling the reference images during few-shot evaluation.

**Evaluation metrics:** In accordance with prior works [10, 18], we report the following evaluation metrics. For AC, we report Area Under the Receiver Operating Characteristics curve (AUROC), Area Under the Precision-Recall curve (AUPR) and F1-score at optimal threshold (F1-max). For AS, we report pixel-wise AUROC (pAUROC), Per-Region Overlap (PRO) and pixel-wise F1-max.

**Implementation details:** For CLIP, we use the OpenCLIP<sup>2</sup> implementation of ViT-B/16+ (240 × 240) trained on LAION-400M [20]. For GEM, we use its official implementation<sup>3</sup>.

## 4.1 Zero-/Few-Shot Anomaly Classification and Segmentation

Tab. 1 compares the AC performance of FADE with two prior work, PatchCore [18] which is an unsupervised anomaly detection method and WinCLIP [10] which is a state-of-the-art zero-/few-shot anomaly detection method. For the zero-shot setting, FADE is a reproduction of WinCLIP without any new additions. For the few-shot setting, FADE outperforms the other methods on all metrics for both MVTec-AD and VisA with a larger improvement seen on VisA. This shows that the vision-guided AS component of FADE is also beneficial to AC.

Tab. 4 compares the AS performance of FADE with PatchCore and WinCLIP. Under zero-shot, FADE significantly outperforms other methods on all metrics for both MVTec-AD and VisA. This is due to the various AS improvements of FADE which include the multi-scale GEM embeddings, a better ChatGPT prompt ensemble and the zero-shot vision-guidance using query image. Under few-shot, FADE performs similar to WinCLIP on MVTec-AD and again outperforms WinCLIP on VisA which is the more challenging dataset. This shows the advantage of FADE on more difficult images. It also shows that vanilla CLIP patch embeddings can be used in place of WinCLIP embeddings for few-shot vision-guided

<sup>2</sup>[https://github.com/mlfoundations/open\\_clip](https://github.com/mlfoundations/open_clip)

<sup>3</sup><https://github.com/WalBouss/GEM>



Anomaly Segmentation		MVTec-AD			VisA		
Setup	Method	pAUROC	PRO	F1-max	pAUROC	PRO	F1-max
0-shot	WinCLIP	85.1±0.0	64.6±0.0	31.7±0.0	79.6±0.0	56.8±0.0	14.8±0.0
	FADE (ours)	<b>89.6±0.0</b>	<b>84.5±0.0</b>	<b>39.8±0.0</b>	<b>91.5±0.0</b>	<b>79.3±0.0</b>	<b>16.7±0.0</b>
1-shot	PatchCore	92.0±1.0	79.7±2.0	50.4±2.1	95.4±0.6	80.5±2.5	38.0±1.9
	WinCLIP+	95.2±0.5	87.1±1.2	<b>55.9±2.7</b>	96.4±0.4	85.1±2.1	41.3±2.3
	FADE (ours)	<b>95.4±0.3</b>	<b>88.3±0.3</b>	54.6±1.1	<b>97.5±0.1</b>	<b>88.9±0.7</b>	<b>42.3±0.5</b>
2-shot	PatchCore	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+	<b>96.0±0.3</b>	88.4±0.9	<b>58.4±1.7</b>	96.8±0.3	86.2±1.4	43.5±3.3
	FADE (ours)	95.8±0.2	<b>88.9±0.2</b>	55.8±1.0	<b>97.8±0.1</b>	<b>89.8±0.4</b>	<b>44.4±0.9</b>
4-shot	PatchCore	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+	<b>96.2±0.3</b>	89.0±0.8	<b>59.5±1.8</b>	97.2±0.2	87.6±0.9	<b>47.0±3.0</b>
	FADE (ours)	<b>96.2±0.1</b>	<b>89.5±0.2</b>	57.0±0.8	<b>98.0±0.0</b>	<b>90.0±0.4</b>	46.4±0.7

Table 4: Comparison of AS performance on MVTec-AD and VisA. We report the mean and standard deviation over 5 random seeds. Bold indicates the best performance.

AC (AUROC)		# shots	
$s^{lang}$	$s^{vis}$	1	4
✓	✗	90.0	90.0
✗	✓	90.7	94.5
✓	✓	<b>93.9</b>	<b>96.3</b>

Table 5: Language- vs vision-guided AC.

AS (pAUROC)		# shots		
$M^{lang}$	$M^{vis}$	0	1	4
✓	✗	86.5	86.5	86.5
✗	✓	86.6	95.1	96.1
✓	✓	<b>89.6</b>	<b>95.4</b>	<b>96.2</b>

Table 6: Language- vs vision-guided AS.

AS. Furthermore, FADE performance has a lower standard deviations across different random runs. This indicates that FADE is more robust to the selection of different reference images. Fig. 1 shows qualitative AS results for some objects. See supplementary for more.

Additional quantitative results comparing FADE and other state-of-the-art methods such as AnomalyCLIP [24], AnomalyGPT [7] and APRIL-GAN [5] can be found in the supplementary. FADE performs competitively even though these other methods require training using additional anomaly detection datasets while FADE does not need any further training.

## 4.2 Ablation Study

**GEM for zero-shot AS:** Tab. 3 shows the impact of using CLIP vs GEM patch embeddings for zero-shot language-guided AS. When CLIP embeddings are used, pAUROC is below 50.0 due to the problem of opposite visualisation between the normal and anomalous regions. See supplementary for qualitative examples of the opposite visualisation when CLIP embeddings are used while GEM embeddings fix this problem. More result comparison on CLIP vs GEM embeddings for the different components of FADE are also shown in the supplementary. The result motivates the choice of CLIP or GEM embeddings used in Fig. 2-a-d.

**Prompt ensemble for zero-shot AS:** Tab. 7 shows the improvement on zero-shot language-guided AS when we use additional text prompts generated automatically by ChatGPT. These prompts capture more diverse descriptions related to the concept of anomaly and normality which allow us to better use the CLIP model for improved AS. Interestingly, we note that the ChatGPT prompts did not improve the zero-shot image-level AC performance.

**Multi-scale aggregation:** Tab. 8 and 9 show the AC and AS performance on MVTec-AD when different input image scale/size is used. Since the patch size is constant (16 pixels), a larger input size means that each patch is covering a finer scale and anomaly with smaller size will be detected and vice versa. Multi-scale refers to taking the average outputs from the three image scales and it has the best results at all k-shot settings since it allows for detecting anomalies of different sizes. See supplementary for qualitative results.

**Language vs vision guidance:** Tab. 5 and 6 show the impact of language- vs vision-guided AC and AS on MVTec-AD. Language guidance depends on text prompts while vision guidance depends on the query (0-shot) or reference images (k-shot). The two types of guidance

Prompt ensemble	MVTec-AD	VisA
WinCLIP prompts	83.7	74.0
WinCLIP + ChatGPT prompts	<b>86.5</b>	<b>87.0</b>

Table 7: Prompt ablations on language-guided AS (pAUROC).

AC (AUROC)	# shots			
	1	2	4	
Image scale	240	92.6	94.4	95.3
	448	92.3	93.7	94.6
	896	90.5	91.5	92.7
Multi-scale	<b>93.9</b>	<b>95.2</b>	<b>96.3</b>	

Table 8: Image scale ablations for AC.

AS (pAUROC)	# shots				
	0	1	2	4	
Image scale	240	87.6	93.4	94.0	94.5
	448	87.4	93.5	94.0	94.5
	896	84.7	91.3	91.7	92.1
Multi-scale	<b>89.6</b>	<b>95.4</b>	<b>95.8</b>	<b>96.2</b>	

Table 9: Image scale ablations for AS.

are complementary and improve the overall performance. See supplementary for qualitative results.

## 5 Conclusion and Future Work

We present a new method FADE that utilises and adapts the CLIP model for zero-/few-shot anomaly detection guided by language and vision. First, we improve language-guided anomaly segmentation by using multi-scale GEM embeddings that are better aligned with language than CLIP embeddings. This is further improved by using an LLM to generate a new prompt ensemble that better captures the concept of anomaly and normality. Finally, vision guidance from the query image further boosts the zero-shot anomaly detection performance while vision guidance from the reference images extends the method to the few-shot setting. On standard benchmarks, FADE performs competitively compared to other state-of-the-art methods with the largest margin of improvement on zero-shot anomaly segmentation.

There are several points about FADE that are worth investigating as future work. First, the use of text prompts generated by ChatGPT is not reproducible. While an alternative open-source LLM can be used to set the seed to ensure reproducibility, it is also worth investigating how sensitive the anomaly detection performance is to the different prompts generated by the same LLM or across different LLMs.

Our experiment results show that for language-guided anomaly detection, GEM embeddings are better for zero-shot AS while CLIP embeddings are more suited for zero-shot AC. On the other hand, for vision-guided anomaly detection, GEM embeddings are better for zero-shot AS while CLIP embeddings are more suited for few-shot AS. A more comprehensive study is needed to gain a better understanding of when to use the GEM and CLIP embeddings under the different detection pipelines and scenarios.

The vision-guided zero-shot AS constructs the memory bank from its own query image patches. This works well at detecting anomalies in textural images (E.g. leather) that normally contain visually similar patches. However, this works less well and produces false positives for anomaly detection in object images (E.g. transistor) whose patches differ to a greater extent. Future work needs to explore other vision-guided zero-shot methods to address this issue.

## Acknowledgement

We would like to thank Romain Sabathe for the insightful discussion and Olivier Koch for the full support throughout this project.

## References

- [1] Paul Bergmann, Michael Fauser, David Sattlegger, and Carsten Steger. MVTec-AD — a comprehensive real-world dataset for unsupervised anomaly detection. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9584–9592, 2019. doi: 10.1109/CVPR.2019.00982.
- [2] 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.
- [3] Walid Bousselham, Felix Petersen, Vittorio Ferrari, and Hilde Kuehne. Grounding everything: Emerging localization properties in vision-language transformers, 2023.
- [4] Yunkang Cao, Xiaohao Xu, Chen Sun, Yuqi Cheng, Zongwei Du, Liang Gao, and Weiming Shen. Segment any anomaly without training via hybrid prompt regularization, 2023.
- [5] Xuhai Chen, Yue Han, and Jiangning Zhang. A zero-/few-shot anomaly classification and segmentation method for CVPR 2023 VAND workshop challenge tracks 1&2: 1st place on zero-shot AD and 4th place on few-shot AD. *arXiv preprint arXiv:2305.17382*, 2023.
- [6] Thomas Defard, Aleksandr Setkov, Angeliqne Loesch, and Romaric Audigier. PaDiM: a patch distribution modeling framework for anomaly detection and localization. In *International Conference on Pattern Recognition*, pages 475–489. Springer, 2021.
- [7] Zhaopeng Gu, Bingke Zhu, Guibo Zhu, Yingying Chen, Ming Tang, and Jinqiao Wang. Anomalygpt: Detecting industrial anomalies using large vision-language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 1932–1940, 2024.
- [8] Denis Gudovskiy, Shun Ishizaka, and Kazuki Kozuka. CFLOW-AD: Real-time unsupervised anomaly detection with localization via conditional normalizing flows. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 98–107, 2022.
- [9] Chaoqin Huang, Haoyan Guan, Aofan Jiang, Ya Zhang, Michael Spratling, and Yan-Feng Wang. Registration based few-shot anomaly detection. In *European Conference on Computer Vision*, pages 303–319. Springer, 2022.
- [10] Jongheon Jeong, Yang Zou, Taewan Kim, Dongqing Zhang, Avinash Ravichandran, and Onkar Dabeer. WinCLIP: Zero-/few-shot anomaly classification and segmentation. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 19606–19616, 2023. doi: 10.1109/CVPR52729.2023.01878.
- [11] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4015–4026, 2023.

- [12] 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.
- [13] Shengze Li, Jianjian Cao, Peng Ye, Yuhan Ding, Chongjun Tu, and Tao Chen. ClipSAM: CLIP and SAM collaboration for zero-shot anomaly segmentation, 2024.
- [14] Xiaofan Li, Zhizhong Zhang, Xin Tan, Chengwei Chen, Yanyun Qu, Yuan Xie, and Lizhuang Ma. PromptAD: Learning prompts with only normal samples for few-shot anomaly detection, 2024.
- [15] Yi Li, Hualiang Wang, Yiqun Duan, and Xiaomeng Li. CLIP surgery for better explainability with enhancement in open-vocabulary tasks, 2023.
- [16] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, et al. Grounding DINO: Marrying dino with grounded pre-training for open-set object detection. *arXiv preprint arXiv:2303.05499*, 2023.
- [17] 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.
- [18] 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.
- [19] Marco Rudolph, Bastian Wandt, and Bodo Rosenhahn. Same same but different: Semi-supervised defect detection with normalizing flows. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 1907–1916, 2021.
- [20] Christoph Schuhmann, Richard Vencu, Romain Beaumont, Robert Kaczmarczyk, Clayton Mullis, Aarush Katta, Theo Coombes, Jenia Jitsev, and Aran Komatsuzaki. LAION-400m: Open dataset of CLIP-filtered 400 million image-text pairs. *arXiv preprint arXiv:2111.02114*, 2021.
- [21] Masato Tamura. Random word data augmentation with CLIP for zero-shot anomaly detection. In *34th British Machine Vision Conference 2023, BMVC 2023, Aberdeen, UK, November 20-24, 2023*. BMVA, 2023. URL <https://papers.bmvc2023.org/0018.pdf>.
- [22] Jih-Ciang Wu, Ding-Jie Chen, Chiou-Shann Fuh, and Tyng-Luh Liu. Learning unsupervised metaformer for anomaly detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4369–4378, 2021.
- [23] Chong Zhou, Chen Change Loy, and Bo Dai. Extract free dense labels from CLIP. In *European Conference on Computer Vision*, pages 696–712. Springer, 2022.

- [24] Qihang Zhou, Guansong Pang, Yu Tian, Shibo He, and Jiming Chen. AnomalyCLIP: Object-agnostic prompt learning for zero-shot anomaly detection. In *The Twelfth International Conference on Learning Representations*, 2024. URL <https://openreview.net/forum?id=buC4E91xZE>.
- [25] Yang Zou, Jongheon Jeong, Latha Pemula, Dongqing Zhang, and Onkar Dabeer. SPot-the-Difference self-supervised pre-training for anomaly detection and segmentation. In *European Conference on Computer Vision*, pages 392–408. Springer, 2022.