Exploring Intrinsic Normal Prototypes within a Single Image for Universal Anomaly Detection

Wei Luo[1*] Yunkang Cao[2*] Haiming Yao[1*] Xiaotian Zhang[1] Jianan Lou[1]
Yuqi Cheng[2] Weiming Shen[2] Wenyong Yu[2†]
[1]Tsinghua University [2]Huazhong University of Science and Technology
{luow23, yhm22, zxt19, ljn22}@mails.tsinghua.edu.cn
{cyk_hust, yuqicheng, shenwm, ywy}@hust.edu.cn

Abstract

Anomaly detection (AD) is essential for industrial inspection, yet existing methods typically rely on "comparing" test images to normal references from a training set. However, variations in appearance and positioning often complicate the alignment of these references with the test image, limiting detection accuracy. We observe that most anomalies manifest as local variations, meaning that even within anomalous images, valuable normal information remains. We argue that this information is useful and may be more aligned with the anomalies since both the anomalies and the normal information originate from the same image. Therefore, rather than relying on external normality from the training set, we propose INP-Former, a novel method that extracts Intrinsic Normal Prototypes (INPs) directly from the test image. Specifically, we introduce the INP Extractor, which linearly combines normal tokens to represent INPs. We further propose an INP Coherence Loss to ensure INPs can faithfully represent normality for the testing image. These INPs then guide the INP-Guided Decoder to reconstruct only normal tokens, with reconstruction errors serving as anomaly scores. Additionally, we propose a Soft Mining Loss to prioritize hard-to-optimize samples during training. INP-Former achieves state-of-the-art performance in single-class, multi-class, and few-shot AD tasks across MVTec-AD, VisA, and Real-IAD, positioning it as a versatile and universal solution for AD. Remarkably, INP-Former also demonstrates some zero-shot AD capability. Code is available at: https://github.com/luow23/INP-Former.

1. Introduction

Unsupervised image anomaly detection (AD) [3, 39] seeks to identify abnormal patterns in images and localize anomalous regions by learning solely from normal samples. This technique has seen widespread application in industrial defect detection [2, 47] and medical disease screening [19]. Recently, various specialized tasks have emerged in response to real-world demands, from conventional single-class AD [28, 34] to more advanced few-shot AD [17, 22] and multi-class AD [14, 40, 42].

Although the composition of normal samples varies across these tasks, the fundamental principle remains unchanged: modeling normality in the training data and assessing whether a test image aligns with this learned normality. However, this approach can be limited due to *misaligned normality* between the training data and the test image. For instance, prototype-based methods [34] extract representative normal prototypes to capture the normality of training samples. In few-shot AD, intra-class variance may lead to poorly aligned prototypes [17], *e.g.*, hazelnuts in different appearances and positions, as shown in Fig. 1(a). Increasing the sample size can mitigate this problem but at the cost of additional prototypes and reduced inference efficiency. When there are multiple classes, *i.e.*, multi-class AD, prototypes from one class may resemble anomalies from another, like the normal background of hazelnut is similar to the anomalies in cable in Fig. 1(b), leading to misclassification.

Several works have focused on extracting normality that is more aligned with the test image. For instance, some studies [13, 17, 45] propose spatially aligning normality within a single class through geometrical transformations. However, spatial alignment is ineffective for certain objects, such as hazelnuts, which exhibit variations beyond spatial positions. Other approaches [27, 40, 41] attempt to divide the normality in the training set into smaller, specific portions and then compare the test image to the corresponding portion of normality, but may still fail to find perfect alignment because of intra-class variances.

Rather than attempting to extract more aligned normal-

*Equal contribution.
†Corresponding author.

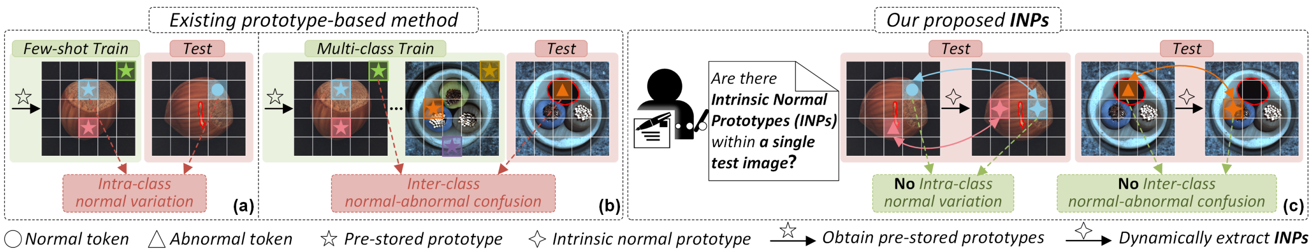

Figure 1. **Motivation for Intrinsic Normal Prototypes (*INPs*).** (a) Pre-stored prototypes from few-shot normal samples may fail to represent all normal patterns. (b) Pre-stored prototypes from one class can be similar to anomalies in another class. (c) The extracted **INPs** are concise yet well-aligned to the test image, alleviating the issues in (a) and (b).

ity from the training set, we propose addressing the issue of misaligned normality by leveraging the *normality within the test image* itself as prototypes, which we term **Intrinsic Normal Prototypes (INPs)**. As illustrated in Fig. 1(c), normal patches within an anomalous test image can function as INPs, and anomalies can be easily detected by comparing them with these INPs. These INPs provide more concise and well-aligned prototypes to the anomalies than those learned from training data, as they typically share the same geometrical context and similar appearances with the abnormal regions within the testing image itself. Accordingly, we explore the prevalence of INPs in various AD scenarios and evaluate their potential to improve AD performance.

Although previous work [1] has attempted to utilize INP for anomaly detection, it employs handcrafted aggregated features as prototypes, thus limiting the method to zero-shot texture anomaly detection. In contrast, we introduce a learnable INP Extractor to extract normal features with adaptable shapes as INPs. We also propose an INP Coherence Loss to ensure that the extracted INPs coherently represent the normality within the test image, avoiding the capture of anomalous regions. However, some weakly representative normal regions are challenging to model with a limited set of discrete INPs, resulting in background noise (Fig. 4(c)). To address this issue, we introduce an INP-Guided Decoder, which integrates INPs into a reconstruction-based framework. This decoder leverages combinations of discrete INPs to accurately reconstruct all normal regions while effectively suppressing the reconstruction of anomalous regions, with reconstruction errors serving as anomaly scores. Furthermore, we introduce a Soft Mining Loss that focuses on normal regions that are challenging to reconstruct, *i.e.*, hard samples, thereby improving overall reconstruction quality and enhancing AD performance.

Our approach, termed INP-Former, primarily leverages vision transformers (ViTs) for both INP extraction and INP-guided reconstruction. Extensive experiments on MVTec-AD [2], VisA [47], and Real-IAD [37] demonstrate that INP-Former achieves superior performance across multi-class, single-class, and few-shot AD tasks, **positioning INP-Former as a universal AD solution**. INP-Former also optimizes computational complexity by extracting concise INPs, *e.g.*, images can be represented effectively using only six INPs, as shown in Sec. 4.3.2. Additionally, as demonstrated in Sec. 4.4.2, INP-Former exhibits strong generalization and can even extract INPs for unseen classes, enabling zero-shot AD capabilities. In summary, our main contributions are:

- We demonstrate that a single image can contain Intrinsic Normal Prototypes (INPs), offering concise and aligned normality for anomaly detection.
- We propose the INP Extractor and incorporate INPs into a reconstruction-based anomaly detection framework using the INP-Guided Decoder.
- We introduce the INP Coherence Loss to extract representative INPs and the Soft Mining Loss to enhance reconstruction quality.

2. Related Works

2.1. Universal Anomaly Detection

There are numerous unsupervised AD tasks, ranging from conventional single-class AD to recent few-shot and multi-class AD setups. We refer to these collectively as universal anomaly detection.

Single-Class Anomaly Detection: This setup was originally introduced by MVTec-AD [2] and involves developing distinct AD models for each class. Typically, images are embedded into a feature space using a pre-trained encoder, after which various schemes, such as reconstruction-based [5, 29, 44], knowledge-distillation-based [9, 36], prototype-based [34, 45], and embedding-based [26, 43] methods, are employed to learn the normality of the given class. While these approaches achieve strong performance, their reliance on class-specific models limits scalability when dealing with a wide range of classes.

Few-shot Anomaly Detection: In practical scenarios, the number of available normal samples may also be limited, motivating the development of few-shot AD methods. In this case, normal samples may not fully capture the variability of normality. To address this challenge, approaches

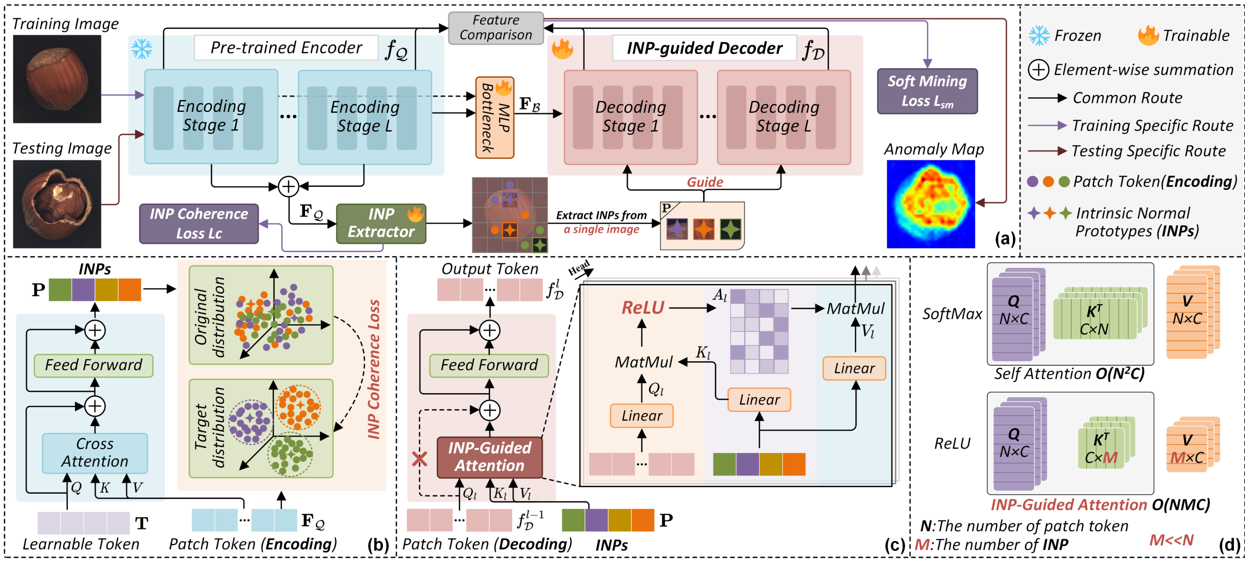

Figure 2. **Overview of our INP-Former framwork for universal anomaly detection.** (a) Our model consists of a pre-trained Encoder, an INP Extractor, a Bottleneck, and an INP-Guided decoder. The INP Extractor dynamically extracts intrinsic normal prototypes from a single image, which the INP-Guided Decoder leverages to effectively suppress anomalous features. (b) Detailed architecture of the INP Extractor. (c) Detailed architecture of each layer in the INP-Guided Decoder. (d) Comparison of computational complexity between INP-Guided Attention and Self Attention. It is important to note that the patch token **(Encoding)** and patch token **(Decoding)** refer to the patch tokens utilized during the encoding and decoding stages, respectively.

such as spatial alignment [17] or contrastive learning [22] are used to create more compact and representative normal embeddings. Recently, Vision-Language Models (VLMs) like CLIP [33] have proven effective for few-shot AD due to their broad, pre-trained knowledge. These VLMs not only provide descriptive visual embeddings but also compute the similarity between text prompts and test images, as seen in works like WinCLIP [21], AnomalyGPT [12], and InCTRL [46]. Some approaches, such as AdaCLIP [4], even enable zero-shot AD through VLMs.

Multi-Class Anomaly Detection: Developing separate models for each class can be resource-intensive, prompting interest in multi-class AD, also known as unified AD [42], which aims to build a single model for multiple classes. UniAD [42] pioneered a unified reconstruction framework for anomaly detection, followed by HVQ-Trans [27] addressed the identical shortcut problem using a vector quantization framework. More recent approaches, such as MambaAD [15] and Dinomaly [14], further enhance multi-class AD performance by leveraging advanced models, *i.e.*, the State Space Model Mamba [11] and DINO [7], respectively. However, these methods lack the functionality to derive aligned normality with the test image. On the contrary, we extract INPs from the testing image, bringing aligned and precise normality for anomaly detection.

2.2. Prototype Learning

Prototype learning [35] aims to extract representative prototypes from a given training set, which are then used for classification by measuring their distances to a test sample in a metric space. This technique is widely used in few-shot learning [23]. Several AD methods also employ prototype learning. For example, PatchCore [34] extracts multiple normal prototypes to represent the normality of the training data, directly computing the minimal distances to the test sample for anomaly detection. Other approaches [10, 18, 30, 32] incorporate prototypes into the reconstruction process to avoid the identical shortcut issue. Specifically, they replace the original inputs with combinations of learned normal prototypes, ensuring that the inputs to the reconstruction model contain only normal elements. However, these methods rely on pre-stored normal prototypes extracted from the training set, which can suffer from the misaligned normality problem. In contrast, our INPs are dynamically extracted from the test image, providing more aligned alternatives for normality representation.

3. Method: INP-Former

3.1. Overview

To fully exploit the advantages of INPs in anomaly detection, we propose INP-Former, as depicted in Fig. 2(a). The model dynamically extracts INPs from a single image and utilizes them to guide the feature reconstruction process, with the reconstruction errors serving as anomaly scores. Specifically, it comprises four key modules: a fixed pre-trained Encoder $\mathcal{Q}$, an INP Extractor $\mathcal{E}$, a Bottleneck $\mathcal{B}$, and an INP-Guided Decoder $\mathcal{D}$. The input image $\mathbf{I} \in \mathbb{R}^{H\times W\times 3}$ is first processed by the pre-trained Encoder $\mathcal{Q}$ to extract multi-scale latent features $f_{\mathcal{Q}} = \{f_{\mathcal{Q}}^1, \dots, f_{\mathcal{Q}}^L | f_{\mathcal{Q}}^l \in \mathbb{R}^{N\times C}, N = \frac{HW}{k^2}\}$, where k represents the downsampling factor. Next, the INP Extractor $\mathcal{E}$ extracts M INPs $\mathbf{P} = \{p_1, \dots, p_M | p_m \in \mathbb{R}^C\}$ from the pre-trained features, with an INP coherence loss ensuring that the extracted INPs consistently represent normal features during testing. The Bottleneck $\mathcal{B}$ subsequently fuses the multi-scale latent features, producing the fused output $F_{\mathcal{B}} = \mathcal{B}(f_{\mathcal{Q}})$. Following the bottleneck, the extracted INPs are utilized to guide the Decoder $\mathcal{D}$ to yield reconstruction outputs $f_{\mathcal{D}} = \{f_{\mathcal{D}}^1, \dots, f_{\mathcal{D}}^L | f_{\mathcal{D}}^l \in \mathbb{R}^{N\times C}\}$ with only normal patterns, thus the reconstruction error between $f_{\mathcal{Q}}$ and $f_{\mathcal{D}}$ can serve as the anomaly score.

3.2. INP Extractor

Existing prototype-based methods [10, 31, 34] store local normal features from the training data and compare them with test images. However, the misaligned normality between these pre-stored prototypes and the test images and the lack of global information lead to suboptimal detection performance. To address these limitations, we propose the INP Extractor to dynamically extract INPs with global information from the test image itself.

Specifically, as illustrated in Fig. 2(b), instead of extracting representative local features as done in PatchCore [34], we employ cross attention to aggregate the global semantic information of the pre-trained features $\mathbf{F}_{\mathcal{Q}} \in \mathbb{R}^{N\times C}$ with M learnable tokens $\mathbf{T} = \{t_1, \dots, t_M | t_m \in \mathbb{R}^C\}$. Here $\mathbf{F}_{\mathcal{Q}}$ is used as the key-value pairs, while $\mathbf{T}$ serve as the query, allowing $\mathbf{T}$ to linearly aggregate $\mathbf{F}_{\mathcal{Q}}$ into INPs $\mathbf{P} = \{p_1, \dots, p_M | p_m \in \mathbb{R}^C\}$.

$$\begin{aligned}
&\mathbf{F}_{\mathcal{Q}} = \mathrm{sum}(\{f_{\mathcal{Q}}^1, \dots, f_{\mathcal{Q}}^L\}) \\
&Q = \mathbf{T}W^Q, K = \mathbf{F}_{\mathcal{Q}}W^K, V = \mathbf{F}_{\mathcal{Q}}W^V \\
&\mathbf{T}' = \mathrm{Attention}(Q, K, V) + \mathbf{T} \\
&\mathbf{P} = \mathrm{FFN}(\mathbf{T}') + \mathbf{T}'
\end{aligned} \tag{1}$$

where $\mathrm{sum}(\cdot)$ denotes the element-wise summation. $Q \in \mathbb{R}^{M\times C}$ and $K, V \in \mathbb{R}^{N\times C}$ represent the query, key and value, respectively. $W^Q, W^k, W^v \in \mathbb{R}^{C\times C}$ are the learnable projection parameters for Q, K, V; $\mathrm{FFN}(\cdot)$ represents the feed-forward network.

To ensure that INPs coherently represent normal features while minimizing the capture of anomalous features during the testing process, we propose an INP coherence loss $\mathcal{L}_c$ to minimize the distances between individual normal features and the corresponding nearest INP.

$$\begin{aligned}
&d_i = \min_{m\in\{1,\dots,M\}} \mathcal{S}(\mathbf{F}_{\mathcal{Q}}(i), p_m) \\
&\mathcal{L}_c = \frac{1}{N}\sum_{i=1}^{N} d_i
\end{aligned} \tag{2}$$

Table 1. Comparison of **computational cost** and **memory usage**.

	Number of multiplicaiton and addition	
Calculation	Vanilla Self Attention	**INP-Guided Attention**
$A_l = Q_l(K_l)^T$	943 496 960	**7 220 640**
$f_{\mathcal{D}}^{l-1'} = A_l V_l$	943 509 504	**6 623 232**
	Memory usage (MB)	
$Q_l/K_l/V_l/A_l$	2.30/2.30/2.30/2.34	2.30/**0.018/0.018/0.018**

where $\mathcal{S}(\cdot, \cdot)$ denotes the cosine distance. d_i represents the distance between the query feature $\mathbf{F}_{\mathcal{Q}}(i)$ and the corresponding nearest INP item. Fig. 4 visually illustrates the effectiveness of $\mathcal{L}_c$.

3.3. INP-Guided Decoder

While we can use the distance between testing features and their nearest INPs for anomaly detection, as illustrated in Fig. 4(c), certain low-representative normal regions are difficult to model with a limited number of discrete INPs, leading to noisy distance maps between these INPs and testing features. To address this issue, we propose the INP-Guided Decoder, aiming to reconstruct these low-representative normal regions through a combination of multiple discrete INPs and suppress the reconstruction of anomalous regions. Additionally, this decoder provides a token-wise discrepancy that can be directly leveraged for anomaly detection. As shown in Fig. 2(c), INPs are incorporated into this decoder to guide the reconstruction process. Since INPs exclusively represent normal patterns in test images, we employ the extracted INPs as key-value pairs, ensuring that the output is a linear combination of normal INPs, thereby effectively suppressing the reconstruction of anomalous queries, *i.e.*, the idenfical mapping issue [42]. Furthermore, we find that the first residual connection can directly introduce anomalous features into the subsequent reconstruction, so we remove this connection in our INP-Guided Decoder. Following the previous work [20], we also employ the ReLU activation function to mitigate the influence of weak correlations and noise on the attention maps.

Mathematically, let $f_{\mathcal{D}}^{l-1} \in \mathbb{R}^{N\times C}$ denotes the output latent features from previous decoding layer. The output $f_{\mathcal{D}}^l \in \mathbb{R}^{N\times C}$ of the l_{th} decoding layer is formulated as,

$$\begin{aligned}
&Q_l = f_{\mathcal{D}}^{l-1}W_l^Q, K_l = \mathbf{P}W_l^K, V_l = \mathbf{P}W_l^V \\
&f_{\mathcal{D}}^{l-1'} = A_l V_l, A_l = \mathrm{ReLU}(Q_l(K_l)^T) \\
&f_{\mathcal{D}}^l = \mathrm{FFN}(f_{\mathcal{D}}^{l-1'}) + f_{\mathcal{D}}^{l-1'}
\end{aligned} \tag{3}$$

where $Q_l \in \mathbb{R}^{N\times C}$ and $K_l, V_l \in \mathbb{R}^{M\times C}$ denote the query, key and value of the l_{th} decoding layer. $W_l^Q, W_l^k, W_l^v \in \mathbb{R}^{C\times C}$ denote the learnable projection parameters for Q_l, K_l, V_l. $A_l \in \mathbb{R}^{N\times M}$ represent the attention map.

Attention Complexity Analysis: As depicted in Fig. 2(d), the computational complexity of vanilla self-attention is

$\mathcal{O}(N^2C)$, while its memory usage is $\mathcal{O}(N^2)$. In contrast, our INP-Guided Attention reduces both the computational complexity and memory usage to $\mathcal{O}(NMC)$ and $\mathcal{O}(NM)$, respectively, which can be approximated as $\mathcal{O}(NC)$ and $\mathcal{O}(N)$ due to $M \ll N$. Tab. 1 offers a detailed comparison of the complexity of vanilla self-attention and INP-Guided Attention. Appendix Sec. C compares the overall complexity between INP-Former and other methods. The light version of INP-Former can even be more efficient than MambaAD [15] yet demonstrate better performance.

3.4. Soft Mining Loss

Inspired by Focal Loss [25], different regions should be assigned varying weights based on their optimization difficulty. Accordingly, we propose Soft Mining Loss to encourage the model to focus more on difficult regions.

Intuitively, the ratio of the reconstruction error of an individual normal region to the average reconstruction error of all normal regions can serve as an indicator of optimization difficulty. Given the encoder $f_{\mathcal{Q}}^l$ and decoder $f_{\mathcal{D}}^l$ features at layer l, let M^l denote the regional cosine distance. Our soft mining loss $\mathcal{L}_{sm}$ is defined as follows:

$$\begin{aligned} w^l(h,w) &= \left[\frac{M^l(h,w)}{u(M^l)}\right]^\gamma \\ \mathcal{L}_{sm} &= \frac{1}{L}\sum_{l=1}^{L} 1 - \frac{vec(f_{\mathcal{Q}}^l)^T \cdot vec(\hat{f}_{\mathcal{D}}^l)}{||vec(f_{\mathcal{Q}}^l)||\,||vec(\hat{f}_{\mathcal{D}}^l)||} \\ \hat{f}_{\mathcal{D}}^l(h,w) &= cg(f_{\mathcal{D}}^l(h,w))_{w^l(h,w)} \end{aligned} \tag{4}$$

where $u(M^l)$ represents the average regional cosine distance within a batch, $\gamma \geq 0$ denotes the temperature hyperparameter, $cg(\cdot)_{w^l(h,w)}$ denotes a gradient adjustment based on dynamic weight $w^l(h,w)$, and $vec(\cdot)$ denotes the flattening operation. The overall training loss of our INP-Former can be expressed as follows: $\mathcal{L}_{total} = \mathcal{L}_{sm} + \lambda \mathcal{L}_c$.

4. Experiments

4.1. Experimental Settings

Datasets: We conduct a comprehensive analysis of the proposed INP-Former on three widely used AD datasets: **MVTec-AD** [2], **VisA** [47], and **Real-IAD** [37]. **MVTec-AD** consists of 15 categories, with 3,629 normal images for training, and 1,982 anomalous images along with 498 normal images for testing. **VisA** contains 12 object categories, with 8,659 normal images for training and 962 normal images along with 1,200 anomalous images for testing. **Real-IAD** includes 30 different objects, with 36,645 normal images for training and 63,256 normal images along with 51,329 anomalous images for testing.

Metrics: Following existing works [14, 15], we use the Area Under the Receiver Operating Characteristic Curve (AUROC), Average Precision (AP), and F1-score-max (F1_max) to evaluate anomaly detection and localization. For anomaly localization specifically, we use Area Under the Per-Region-Overlap (AUPRO) as an additional metric.

Implementation Details: INP-Former adopts ViT-Base/14 with DINO2-R [7] weights as the default pre-trained encoder. The number M of INPs is set to six by default. The INP Extractor includes a standard Vision Transformer block. The layer number of the INP-Guided decoder is eight. All input images are resized to 448^2 and then center-cropped to 392^2. The hyperparameters γ and λ are set to 3.0 and 0.2, respectively. We use the StableAdamW [38] optimizer with a learning rate $1e^{-3}$ and a weight decay of $1e^{-4}$ for 200 epochs. Notably, the above hyperparameters do not require any adjustment across the three datasets. Appendix Sec. A presents more implementation details. Appendix Sec. D, E, and F analyze the influence of input resolution, ViT architecture, and λ, respectively.

4.2. Main Results

4.2.1 Multi-Class Anomaly Detection

We compare the proposed INP-Former with several state-of-the-art (SOTA) methods for multi-class anomaly detection, including reconstruction-based methods RD4AD [9], UniAD [42], DiAD [16], MambaAD [15], and Dinomaly [14], and embedding-based methods SimpleNet [26] and DeSTSeg [43]. A detailed introduction to the comparison methods can be found in Appendix Sec. B.

The experimental results on the three AD datasets are presented in Tab. 2. On the widely used MVTec-AD dataset, our method achieves SOTA performance, with image-level metrics of **99.7/99.9/99.2** and pixel-level metrics of **98.5/71.0/69.7/94.9**. On VisA, our method achieves competitive results, attaining the best image-level metrics of **98.9/99.0/96.6**, and achieving the best or second-best pixel-level performance of **98.8**/<u>51.2</u>/<u>54.7</u>/<u>94.4</u>. On the more complex and challenging Real-IAD dataset, our method reaches new SOTA performance, with image-level metrics of **90.5/88.1/81.5** and pixel-level metrics of **99.0/47.5/50.3/95.0**. Compared to the second-best results, our method improves by 1.2↑/1.3↑/1.3↑ at the image level and by 0.2↑/4.7↑/3.2↑/1.1↑ at the pixel level. The SOTA performance achieved across the three datasets showcases the effectiveness and robustness of our method. The per-class performance metrics are presented in Appendix Sec. H. Appendix Sec. G presents the performance of INP-Former in a more challenging scenario, which we call super-multi-class anomaly detection, thus training INP-Former on several datasets – MVTec-AD, VisA, and Real-IAD – simultaneously. Results show that our method can even detect anomalies in more classes without significant performance degradation. Fig. 3 demonstrates the precise anomaly localization capability of our method. More qualitative results

Table 2. **Multi-class** anomaly detection performance on different AD datasets. The best in **bold**, the second-highest is underlined.

Dataset →	MVTec-AD [2]		VisA [47]		Real-IAD [37]	
Metric →	Image-level(I-AUROC/I-AP/I-F1_max)			Pixel-level(P-AUROC/P-AP/P-F1_max/AUPRO)		
Method ↓	Image-level	Pixel-level	Image-level	Pixel-level	Image-level	Pixel-level
RD4AD [9]	94.6/96.5/95.2	96.1/48.6/53.8/91.1	92.4/92.4/89.6	98.1/38.0/42.6/91.8	82.4/79.0/73.9	97.3/25.0/32.7/89.6
UniAD [42]	96.5/98.8/96.2	96.8/43.4/49.5/90.7	88.8/90.8/85.8	98.3/33.7/39.0/85.5	83.0/80.9/74.3	97.3/21.1/29.2/86.7
SimpleNet [26]	95.3/98.4/95.8	96.9/45.9/49.7/86.5	87.2/87.0/81.8	96.8/34.7/37.8/81.4	57.2/53.4/61.5	75.7/2.8/6.5/39.0
DeSTSeg [43]	89.2/95.5/91.6	93.1/54.3/50.9/64.8	88.9/89.0/85.2	96.1/39.6/43.4/67.4	82.3/79.2/73.2	94.6/37.9/41.7/40.6
DiAD [16]	97.2/99.0/96.5	96.8/52.6/55.5/90.7	86.8/88.3/85.1	96.0/26.1/33.0/75.2	75.6/66.4/69.9	88.0/2.9/7.1/58.1
MambaAD [15]	98.6/99.6/97.8	97.7/56.3/59.2/93.1	94.3/94.5/89.4	98.5/39.4/44.0/91.0	86.3/84.6/77.0	98.5/33.0/38.7/90.5
Dinomaly [14]	99.6/99.8/99.0	98.4/69.3/69.2/94.8	98.7/98.9/96.2	98.7/**53.2**/**55.7**/**94.5**	89.3/86.8/80.2	98.8/42.8/47.1/93.9
INP-Former	**99.7/99.9/99.2**	**98.5/71.0/69.7/94.9**	**98.9/99.0/96.6**	**98.9**/51.2/54.7/94.4	**90.5/88.1/81.5**	**99.0/47.5/50.3/95.0**

Table 3. **Few-shot (4-shot)** anomaly detection performance on different AD datasets. The best in **bold**, the second-highest is underlined. † indicates the results we reproduced using publicly available code.

Dataset →	MVTec-AD [2]		VisA [47]		Real-IAD [37]	
Method ↓	Image-level	Pixel-level	Image-level	Pixel-level	Image-level	Pixel-level
SPADE [6]	84.8/92.5/91.5	92.7/-/46.2/87.0	81.7/83.4/82.1	96.6/-/43.6/87.3	50.8†/45.8†/61.2†	59.5†/0.2†/0.5†/19.2†
PaDiM [8]	80.4/90.5/90.2	92.6/-/46.1/81.3	72.8/75.6/78.0	93.2/-/24.6/72.6	60.3†/53.5†/64.0†	90.9†/2.1†/5.1†/67.6†
PatchCore [34]	88.8/94.5/92.6	94.3/-/55.0/84.3	85.3/87.5/84.3	96.8/-/43.9/84.9	66.0†/62.2†/65.2†	92.9†/9.8†/16.1†/68.6†
WinCLIP [21]	95.2/97.3/94.7	96.2/-/59.5/89.0	87.3/88.8/84.2	97.2/-/47.0/87.6	73.0†/61.8†/61.0†	93.8†/13.3†/21.0†/76.4†
PromptAD [24]	96.6/-/-	96.5/-/-/90.5	89.1/-/-	97.4/-/-/86.2	59.7†/43.5†/52.9†	86.9†/8.7†/16.1†/61.9†
INP-Former	**97.6/98.6/97.0**	**97.0/65.9/65.6/92.9**	**96.4/96.0/93.0**	**97.7/49.3/54.3/93.1**	**76.7/72.3/71.7**	**97.3/32.2/36.7/89.0**

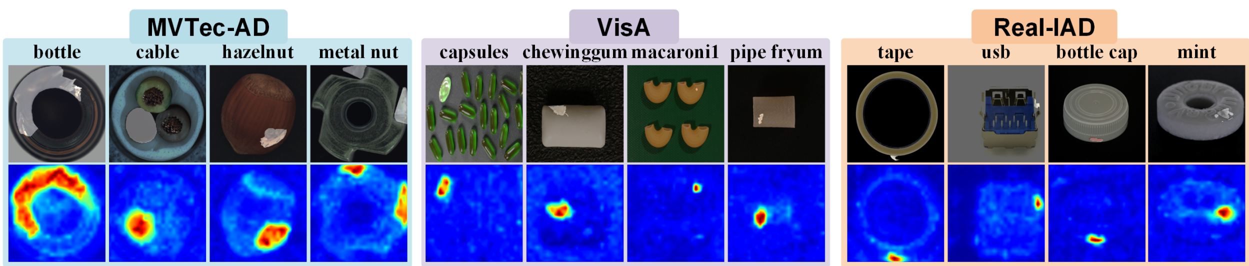

Figure 3. **Qualitative results of anomaly localization** on the MVTec-AD [2], VisA [47], and Real-IAD [37] datasets for **multi-class anomaly detection**. The first row presents the input images with their ground truth, while the second row displays the corresponding anomaly maps.

are presented in Appendix Sec. L.

4.2.2 Few-Shot Anomaly Detection

We compare our method with several SOTA approaches for few-shot anomaly detection, including prototype-based methods SPADE [6], PaDiM [8], and PatchCore [34] and recent advances that utilize VLMs, *i.e.*, WinCLIP [21] and PromptAD [24].

As shown in Tab. 3, our method significantly outperforms previous SOTAs on three different AD datasets. Compared to the second-best results, our method achieves improvements of 1.0↑/1.3↑/2.3↑ in image-level scores and 0.5↑/-/6.1↑/2.4↑ in pixel-level scores on the MVTec-AD dataset. It similarly outperforms the second-best results on the VisA dataset, with enhancements of 7.3↑/7.2↑/8.7↑ for image-level and 0.3↑/-/7.3↑/5.5↑ for pixel-level scores. Additionally, on the Real-IAD dataset, our method surpasses the second-best results by 3.7↑/10.1↑/6.5↑ in image-level and 3.5↑/18.9↑/15.7↑/12.6↑ in pixel-level scores. The superior performance of our method in few-shot anomaly detection stems from its ability to extract INPs from a single image, eliminating the need for extensive normal data to pre-store prototypes. More comparison results on 1-shot and 2-shot are presented in Appendix Sec. I.

4.2.3 Single-Class Anomaly Detection

We further compared our proposed INP-Former with current SOTA methods for single-class anomaly detection, as shown in Tab. 4. The results indicate that INP-Former

Table 4. **Single class** anomaly detection performance on different AD datasets. The best in **bold**, the second-highest is <u>underlined</u>.

Dataset →	MVTec-AD [2]			VisA [47]			Real-IAD [37]		
Method ↓	I-AUROC	P-AP	AUPRO	I-AUROC	P-AP	AUPRO	I-AUROC	P-AP	AUPRO
PatchCore [34]	99.1	56.1	93.5	95.1	40.1	91.2	89.4	-	91.5
RD4AD [9]	98.5	58.0	93.9	96.0	27.7	70.9	87.1	-	93.8
SimpleNet [26]	<u>99.6</u>	54.8	90.0	96.8	36.3	88.7	88.5	-	84.6
Dinomaly [14]	**99.7**	<u>68.9</u>	<u>95.0</u>	**98.9**	**50.7**	**95.1**	<u>92.0</u>	<u>45.2</u>	<u>95.1</u>
INP-Former	**99.7**	**70.2**	**95.4**	<u>98.5</u>	<u>49.2</u>	<u>93.8</u>	**92.1**	**48.1**	**95.6**

Table 5. **Overall ablation** on MVTec-AD [2] and VisA [47] datasets. ***"INP"*** refers to the use of INP Extractor and INP-Guided Decoder.

	Dataset →			MVTec-AD [2]		VisA [47]	
	"INP"	$\mathcal{L}_c$	$\mathcal{L}_{sm}$	Image-level	Pixel-level	Image-level	Pixel-level
Module	✘	✘	✘	98.59/99.18/97.63	97.19/61.73/62.94/92.73	96.58/97.18/92.89	97.50/47.24/51.90/82.85
	✔	✘	✘	99.53/99.80/98.81	98.32/69.82/69.38/94.69	98.11/98.23/95.22	98.41/50.34/54.23/93.63
	✔	✔	✘	99.61/99.83/99.02	98.39/70.01/69.53/**95.10**	98.16/98.30/95.47	98.46/51.09/54.46/93.71
	✔	✔	✔	**99.67/99.88/99.20**	**98.48/71.02/69.65**/94.87	**98.90/99.02/96.57**	**98.90/51.22/54.74/94.36**

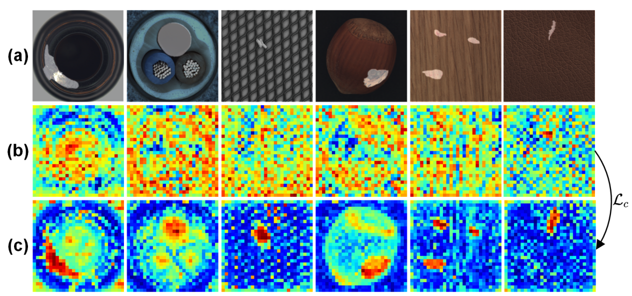

Figure 4. **Visualization of the impact of INP coherence loss** $\mathcal{L}_c$. (a) Input anomalous image and ground truth. (b) Distance map **without** $\mathcal{L}_c$. (c) Distance map **with** $\mathcal{L}_c$. The distance map is obtained by calculating the distance between the input features and their nearest INP terms, as described in Eq. 2.

achieves new SOTA performance on the MVTec-AD and Real-IAD datasets and demonstrates competitive performance on the VisA dataset. Per-category performance of INP-Former is presented in Appendix Sec. J.

4.3. Ablation Study

4.3.1 Overall Ablation

As shown in Tab. 5, we conduct comprehensive experiments on MVTec-AD [2] and VisA [47] to validate the effectiveness of the proposed components, *i.e.*, INP Extractor and INP-Guided Decoder (***"INP"***), INP Coherence Loss ($\mathcal{L}_c$), and Soft Mining Loss ($\mathcal{L}_{sm}$). In the first row, we train a baseline model without incorporating any proposed module, similar to the RD4AD [9] framework. The results in the second row demonstrate that ***"INP"*** significantly enhances overall performance. This improvement arises from the fact that ***"INP"*** introduces an information bottleneck, which effectively helps the model preserve normal features while filtering out anomalous ones. The results in the third row

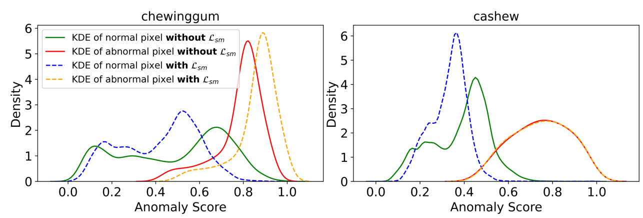

Figure 5. **Visualization of the impact of soft mining loss** $\mathcal{L}_{sm}$. We plot the Kernel Density Estimation (KDE) for the chewinggum and cashew categories in the VisA [47] dataset to estimate the probability density of the anomaly scores.

indicate that $\mathcal{L}_c$ enhances the model's performance. This improvement stems from $\mathcal{L}_c$ ensuring that the extracted INP coherently represents normal patterns, thereby avoiding the capture of anomalous ones and establishing a solid foundation for the subsequent suppression of anomalous feature reconstruction. Fig. 4 provides a more intuitive demonstration of the effectiveness of $\mathcal{L}_c$. The last row indicates that $\mathcal{L}_{sm}$ boosts overall performance, as $\mathcal{L}_{sm}$ directs the model's attention toward more challenging regions, thereby unlocking its optimal performance. Fig. 5 visually illustrates the impact of $\mathcal{L}_{sm}$, from which we can see that $\mathcal{L}_{sm}$ contributes to a smaller overlap between anomaly score distributions of normal and abnormal pixels.

4.3.2 Influence of the Number of INPs

As shown in Fig. 6, we conduct an ablation analysis on the number M of INPs. The experimental results indicate that when M exceeds four, the model's performance stabilizes. However, if M becomes excessively large, the extracted INPs may also comprise information from abnormal tokens, leading to a slight decline in overall performance. In our study, we set M to six.

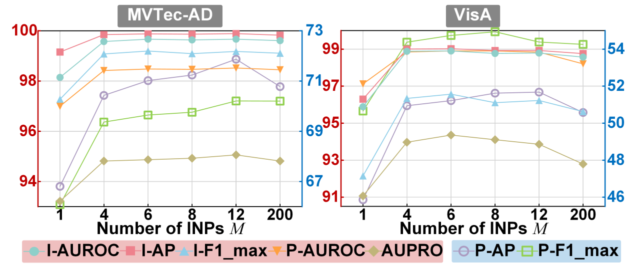

Figure 6. **Influence of the number of INPs** M on model performance across the MVTec-AD [2] and VisA [47] datasets. Pixel-level AP and F1_max use the right vertical axis, while the other metrics share the left vertical axis.

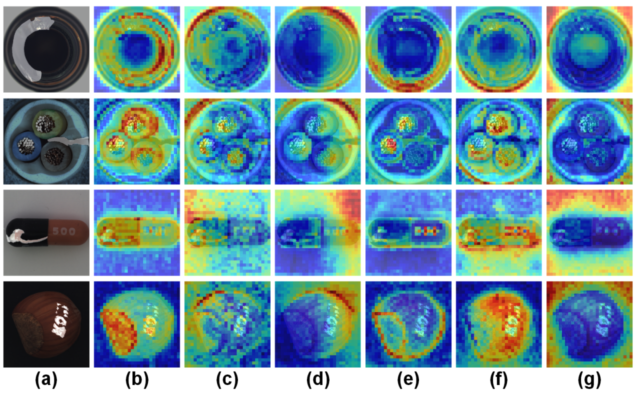

Figure 7. **Visualizations of INPs**. (a) Input anomalous image and ground truth. (b)-(g) Attention maps of six different INPs.

4.4. Exploration on INPs

4.4.1 Visualizations of INPs

As shown in Fig. 7, INPs effectively capture different semantic information. Specifically, the learned INPs focus on various regions of the image, including object areas (Fig. 7(b), (e) and (f)), object edges (Fig. 7(c) and (d)), and background areas (Fig. 7(g)). This diversity is attributed to our design of guiding the reconstruction process with INPs. Additionally, INP coherence loss ensures consistency in representing normal features, allowing INPs to concentrate solely on normal regions while ignoring anomalies. This mechanism ensures the decoder reconstructs features containing only normal patterns, thereby improving anomaly detection performance.

4.4.2 Generalization capabilities of INP Extractor

As shown in Fig. 8, the INP Extractor trained on the Real-IAD [37] dataset is capable of detecting INPs on the unseen MVTec-AD [2] dataset, and the distance maps to these INPs can serve for zero-shot anomaly detection. This effectively demonstrates the INP Extractor's ability to dynamically extract INPs from a single image, with the INP coherence loss $\mathcal{L}_c$ ensuring that the extracted INPs coherently capture normal patterns. As shown in Appendix Sec. K, without any specific training for zero-shot anomaly detection, our method can even outperform a specified method WinCLIP [21], achieving 88.0 and 88.7 pixel-level AUROCs on MVTec-AD and VisA, respectively.

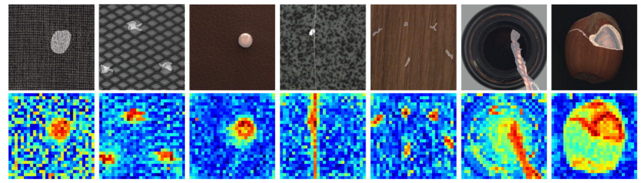

Figure 8. **Zero-shot anomaly detection results**. Here INP-Former is trained on Real-IAD [37] and tested on MVTec-AD [2]. Distance maps are visualized.

5. Conclusion

We propose INP-Former, a novel method for anomaly detection that explores the role of INPs. By learning to linearly combine normal tokens into INPs and using these INPs to guide the reconstruction of normal tokens, INP-Former significantly enhances anomaly detection performance. The introduction of the INP Coherence Loss and Soft Mining Loss further refines INP quality and optimizes the training process. Extensive experiments on MVTec-AD, VisA, and Real-IAD datasets demonstrate that INP-Former achieves SOTA or comparable performance across single-class, multi-class, and few-shot anomaly detection tasks. These results validate the existence and effectiveness of INPs, which can even be extracted from images in unseen categories, enabling zero-shot anomaly detection.

Limitations & Future Works. Our method encounters certain limitations when detecting logical anomalies that closely resemble the background distribution, such as the misplaced anomalies in the Transistor class of the MVTec-AD dataset. This issue primarily arises because the misplaced anomaly in Transistor is highly similar to the background, causing INP Extractor to incorrectly extract this anomaly as INPs. A more detailed discussion can be found in Appendix Sec. M. In future work, we plan to combine the proposed INPs with pre-stored prototypes to address this limitation. While the pre-stored prototypes encapsulate comprehensive semantic information, the INPs exhibit strong alignment. This integration is expected to significantly improve the model's ability to detect logical anomalies that are similar to the background.

Acknowledge

This study was supported by the National Natural Science Foundation of China under Grant No. 52375494.

References

[1] Toshimichi Aota, Lloyd Teh Tzer Tong, and Takayuki Okatani. Zero-shot versus many-shot: Unsupervised texture anomaly detection. In *2023 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 5553–5561, 2023. 2

[2] Paul Bergmann, Kilian Batzner, Michael Fauser, David Sattlegger, and Carsten Steger. The MVTec anomaly detection dataset: A comprehensive real-world dataset for unsupervised anomaly detection. *International Journal of Computer Vision*, 129(4):1038–1059, 2021. 1, 2, 5, 6, 7, 8

[3] Yunkang Cao, Xiaohao Xu, Jiangning Zhang, Yuqi Cheng, Xiaonan Huang, Guansong Pang, and Weiming Shen. A survey on visual anomaly detection: Challenge, approach, and prospect. *arXiv preprint arXiv:2401.16402*, 2024. 1

[4] Yunkang Cao, Jiangning Zhang, Luca Frittoli, Yuqi Cheng, Weiming Shen, and Giacomo Boracchi. Adaclip: Adapting clip with hybrid learnable prompts for zero-shot anomaly detection. In *European Conference on Computer Vision*. Springer, 2024. 3

[5] Yunkang Cao, Haiming Yao, Wei Luo, and Weiming Shen. Varad: Lightweight high-resolution image anomaly detection via visual autoregressive modeling. *IEEE Transactions on Industrial Informatics*, pages 1–10, 2025. 2

[6] Niv Cohen and Yedid Hoshen. Sub-image anomaly detection with deep pyramid correspondences. *arXiv preprint arXiv:2005.02357*, 2020. 6

[7] Timothée Darcet, Maxime Oquab, Julien Mairal, and Piotr Bojanowski. Vision transformers need registers. *arXiv preprint arXiv:2309.16588*, 2023. 3, 5

[8] Thomas Defard, Aleksandr Setkov, Angelique Loesch, and Romaric Audigier. Padim: a patch distribution modeling framework for anomaly detection and localization. In *International Conference on Pattern Recognition*, pages 475–489. Springer, 2021. 6

[9] Hanqiu Deng and Xingyu Li. Anomaly detection via reverse distillation from one-class embedding. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9737–9746, 2022. 2, 5, 6, 7

[10] Dong Gong, Lingqiao Liu, Vuong Le, Budhaditya Saha, Moussa Reda Mansour, Svetha Venkatesh, and Anton van den Hengel. Memorizing normality to detect anomaly: Memory-augmented deep autoencoder for unsupervised anomaly detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019. 3, 4

[11] Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023. 3

[12] 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*, pages 1932–1940, 20 24. 3

[13] Hewei Guo, Liping Ren, Jingjing Fu, Yuwang Wang, Zhizheng Zhang, Cuiling Lan, Haoqian Wang, and Xinwen Hou. Template-guided hierarchical feature restoration for anomaly detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6447–6458, 2023. 1

[14] Jia Guo, Shuai Lu, Weihang Zhang, Fang Chen, Hongen Liao, and Huiqi Li. Dinomaly: The less is more philosophy in multi-class unsupervised anomaly detection. *arXiv preprint arXiv:2405.14325*, 2024. 1, 3, 5, 6, 7

[15] Haoyang He, Yuhu Bai, Jiangning Zhang, Qingdong He, Hongxu Chen, Zhenye Gan, Chengjie Wang, Xiangtai Li, Guanzhong Tian, and Lei Xie. MambaAD: Exploring state space models for multi-class unsupervised anomaly detection. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. 3, 5, 6

[16] Haoyang He, Jiangning Zhang, Hongxu Chen, Xuhai Chen, Zhishan Li, Xu Chen, Yabiao Wang, Chengjie Wang, and Lei Xie. A diffusion-based framework for multi-class anomaly detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 8472–8480, 2024. 5, 6

[17] 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. 1, 3

[18] Chao Huang, Chengliang Liu, Zheng Zhang, Zhihao Wu, Jie Wen, Qiuping Jiang, and Yong Xu. Pixel-level anomaly detection via uncertainty-aware prototypical transformer. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 521–530, 2022. 3

[19] Chaoqin Huang, Aofan Jiang, Jinghao Feng, Ya Zhang, Xinchao Wang, and Yanfeng Wang. Adapting visual-language models for generalizable anomaly detection in medical images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11375–11385, 2024. 1

[20] Wenli Huang, Ye Deng, Siqi Hui, Yang Wu, Sanping Zhou, and Jinjun Wang. Sparse self-attention transformer for image inpainting. *Pattern Recognition*, 145:109897, 2024. 4

[21] Jongheon Jeong, Yang Zou, Taewan Kim, Dongqing Zhang, Avinash Ravichandran, and Onkar Dabeer. Winclip: Zero-/few-shot anomaly classification and segmentation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 19606–19616, 2023. 3, 6, 8

[22] Yuxin Jiang, Yunkang Cao, and Weiming Shen. Prototypical learning guided context-aware segmentation network for few-shot anomaly detection. *IEEE Transactions on Neural Networks and Learning Systems*, pages 1–11, 2024. 1, 3

[23] Gen Li, Varun Jampani, Laura Sevilla-Lara, Deqing Sun, Jonghyun Kim, and Joongkyu Kim. Adaptive prototype learning and allocation for few-shot segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8334–8343, 2021. 3

[24] 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. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16838–16848, 2024. 6

[25] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *2017*

IEEE International Conference on Computer Vision (ICCV), pages 2999–3007, 2017. 5

[26] Zhikang Liu, Yiming Zhou, Yuansheng Xu, and Zilei Wang. Simplenet: A simple network for image anomaly detection and localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20402–20411, 2023. 2, 5, 6, 7

[27] Ruiying Lu, YuJie Wu, Long Tian, Dongsheng Wang, Bo Chen, Xiyang Liu, and Ruimin Hu. Hierarchical vector quantized transformer for multi-class unsupervised anomaly detection. *Advances in Neural Information Processing Systems*, 36:8487–8500, 2023. 1, 3

[28] Wei Luo, Haiming Yao, and Wenyong Yu. Template-based feature aggregation network for industrial anomaly detection. *Engineering Applications of Artificial Intelligence*, 131: 107810, 2024. 1

[29] Wei Luo, Haiming Yao, Wenyong Yu, and Zhengyong Li. AMI-Net: Adaptive mask inpainting network for industrial anomaly detection and localization. *IEEE Transactions on Automation Science and Engineering*, 2024. 2

[30] Hui Lv, Chen Chen, Zhen Cui, Chunyan Xu, Yong Li, and Jian Yang. Learning normal dynamics in videos with meta prototype network. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 15425–15434, 2021. 3

[31] Hyunjong Park, Jongyoun Noh, and Bumsub Ham. Learning memory-guided normality for anomaly detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 4

[32] Hyunjong Park, Jongyoun Noh, and Bumsub Ham. Learning memory-guided normality for anomaly detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14372–14381, 2020. 3

[33] 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, 2021. 3

[34] Karsten Roth, Latha Pemula, Joaquin Zepeda, Bernhard Schölkopf, Thomas Brox, and Peter Gehler. Towards total recall in industrial anomaly detection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14318–14328, 2022. 1, 2, 3, 4, 6, 7

[35] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. *Advances in neural information processing systems*, 30, 2017. 3

[36] Tran Dinh Tien, Anh Tuan Nguyen, Nguyen Hoang Tran, Ta Duc Huy, Soan T.M. Duong, Chanh D. Tr. Nguyen, and Steven Q. H. Truong. Revisiting reverse distillation for anomaly detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 24511–24520, 2023. 2

[37] Chengjie Wang, Wenbing Zhu, Bin-Bin Gao, Zhenye Gan, Jiangning Zhang, Zhihao Gu, Shuguang Qian, Mingang Chen, and Lizhuang Ma. Real-iad: A real-world multi-view dataset for benchmarking versatile industrial anomaly detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22883–22892, 2024. 2, 5, 6, 7, 8

[38] Mitchell Wortsman, Tim Dettmers, Luke Zettlemoyer, Ari Morcos, Ali Farhadi, and Ludwig Schmidt. Stable and low-precision training for large-scale vision-language models. *Advances in Neural Information Processing Systems*, 36: 10271–10298, 2023. 5

[39] Guoyang Xie, Jinbao Wang, Jiaqi Liu, Jiayi Lyu, Yong Liu, Chengjie Wang, Feng Zheng, and Yaochu Jin. IM-IAD: Industrial image anomaly detection benchmark in manufacturing. *IEEE Transactions on Cybernetics*, pages 1–14, 2024. 1

[40] Haiming Yao, Yunkang Cao, Wei Luo, Weihang Zhang, Wenyong Yu, and Weiming Shen. Prior normality prompt transformer for multiclass industrial image anomaly detection. *IEEE Transactions on Industrial Informatics*, 20(10): 11866–11876, 2024. 1

[41] Xincheng Yao, Ruoqi Li, Zefeng Qian, Lu Wang, and Chongyang Zhang. Hierarchical gaussian mixture normalizing flow modeling for unified anomaly detection. *arXiv preprint arXiv:2403.13349*, 2024. 1

[42] Zhiyuan You, Lei Cui, Yujun Shen, Kai Yang, Xin Lu, Yu Zheng, and Xinyi Le. A unified model for multi-class anomaly detection. In *Advances in Neural Information Processing Systems*, pages 4571–4584, 2022. 1, 3, 4, 5, 6

[43] Xuan Zhang, Shiyu Li, Xi Li, Ping Huang, Jiulong Shan, and Ting Chen. Destseg: Segmentation guided denoising student-teacher for anomaly detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3914–3923, 2023. 2, 5, 6

[44] Ximiao Zhang, Min Xu, and Xiuzhuang Zhou. Realnet: A feature selection network with realistic synthetic anomaly for anomaly detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16699–16708, 2024. 2

[45] Ye Zheng, Xiang Wang, Rui Deng, Tianpeng Bao, Rui Zhao, and Liwei Wu. Focus your distribution: Coarse-to-fine non-contrastive learning for anomaly detection and localization. In *2022 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6. IEEE, 2022. 1, 2

[46] Jiawen Zhu and Guansong Pang. Toward generalist anomaly detection via in-context residual learning with few-shot sample prompts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17826–17836, 2024. 3

[47] Yang Zou, Jongheon Jeong, Latha Pemula, Dongqing Zhang, and Onkar Dabeer. Spot-the-difference self-supervised pretraining for anomaly detection and segmentation. In *European Conference on Computer Vision*, pages 392–408. Springer, 2022. 1, 2, 5, 6, 7, 8