Explainable Visual Anomaly Detection via Concept Bottleneck Models

Arianna Stropeni
arianna.stropeni@studenti.unipd.it
Valentina Zaccaria
valentina.zaccaria@unipd.it
Francesco Borsatti
francesco.borsatti.1@phd.unipd.it
Davide Dalle Pezze
davide.dallepezze@unipd.it
Manuel Barusco
manuel.barusco@phd.unipd.it
Gian Antonio Susto
gianantonio.susto@unipd.it

Department of Information Engineering
University of Padova
Padova, Italy

Abstract

In recent years, Visual Anomaly Detection (VAD) has attracted significant attention due to its high interpretability, as it provides visual explanations by highlighting anomalous regions within an image. However, although these visual explanations can be helpful, they lack a direct and semantically meaningful interpretation for users. To address this limitation, we propose extending Concept Bottleneck Models (CBMs) to the VAD setting. By learning meaningful concepts, the network can provide human-interpretable descriptions of anomalies, offering a novel and more insightful way to explain them. Our main contributions are: (i) we introduce a concept-based framework for anomaly explanation by extending CBMs to the VAD setting for the first time; (ii) we evaluate multiple supervision regimes, ranging from supervised to unsupervised anomaly settings; (iii) we propose a dual-branch architecture that combines a CBM-based Concept branch for concept-level explanations with a Visual branch for pixel-level anomaly localization, bridging semantic and spatial interpretability. When evaluated across three well-established VAD benchmarks, our approach, Concept-Aware Visual Anomaly Detection (CONVAD), achieves performance comparable to classic VAD methods (up to 0.97 I-AUROC on MVTec AD) while providing richer, concept-driven explanations. A user study (582 responses) shows that CONVAD explanations are rated effective in 73.5% of cases (vs. 60.7% for the SOTA VLM-based explainer Anomaly-OV) and misleading in only 3.1% (vs. 11.7%). Beyond explanation quality, CBMs enable human-in-the-loop correction: concept-level interventions further boost detection in the challenging unsupervised setting, closing part of the gap to fully supervised performance.

1 Introduction

Visual Anomaly Detection (VAD) aims to detect anomalies in images while highlighting the pixels responsible for the anomaly. These capabilities are particularly relevant in domains

© 2026. The copyright of this document resides with its authors.
It may be distributed unchanged freely in print or electronic forms.

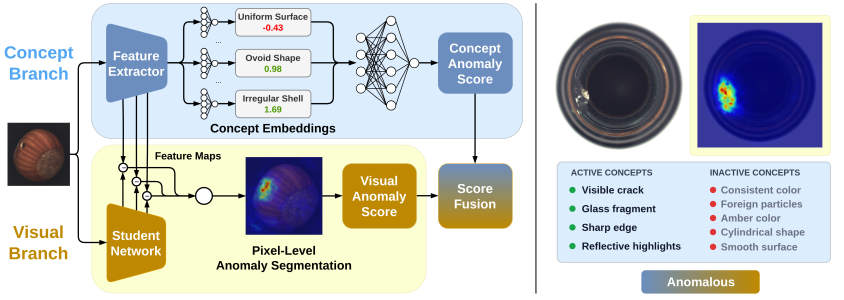


Figure 1: (Left) CONVAD Architecture, (Right) Output example of CONVAD: the input image, the anomaly heatmap, a list of most active and inactive concepts, and the prediction.

such as manufacturing, medicine, and surveillance [2, 6, 8].

Although VAD models can produce anomaly heatmaps, it is questionable whether these visual localizations alone are sufficient for interpretability. Indeed, identifying where an anomaly occurs is sometimes insufficient to understand what is actually anomalous, since this depends on the context and on the underlying notion of normality. In this work, we address this limitation by complementing spatial anomaly heatmaps with semantic explanations, called *concepts*, by considering a different VAD paradigm based on Concept Bottleneck Models [17] (CBMs). Unlike standard VAD approaches, CBMs rely on supervised concept annotations and explicitly learn intermediate, human-interpretable concepts that drive the final prediction. Incorporating CBMs into the VAD framework enables the system to provide not only image- and pixel-level labels but also a set of interpretable concepts underlying each prediction, which can be used by the end user to understand model outputs and enhance the decision-making process. In addition, CBMs allow users to intervene on intermediate concept activations, aiding the model in making correct predictions and enabling seamless **human-machine collaboration**, which is not possible in standard VAD models.

A potential limitation of CBMs is that they require supervised training. This differs from the traditional VAD setting which assumes only normal samples available, since collecting and annotating anomalous examples is expensive, especially in industrial settings. Despite this, a growing body of recent studies has shown that incorporating limited labeled information into VAD can substantially improve anomaly detection performance [25, 34]. Such supervision may take the form of weak labels or a few real anomalous samples. Overall, these findings suggest that a balanced trade-off between annotation effort and model capability is both practical and beneficial. Following this direction, our approach adopts a supervised setting under limited supervision, considering scenarios with either very few real anomalies or synthetic anomalies. This minimal annotation effort is compensated by gains in interpretability and more effective human-collaboration.

Our contributions can be summarized as follows:

(i) **Adaptation of CBMs to VAD.** To the best of our knowledge, this work is the first to successfully adapt CBMs to the VAD scenario. This adaptation is non-trivial and requires addressing several challenges. CBMs require a *Concept Dataset* for training, where each sample is annotated with concepts from a predefined vocabulary. Since no such dataset exists for VAD, to conduct an extensive evaluation of our method, we adopt an automated, VLM-based pipeline for extracting meaningful concepts and annotating industrial images. This is an existing approach for CBMs literature in other domains [21, 33, 43]. We publicly release

the resulting Concept Datasets for three major industrial VAD benchmarks (MVTec AD, VisA, and Real-IAD) to facilitate future research, along with all code (see Supplementary Material) used for the experiments.

(ii) **Evaluation of supervision regimes** and synthetic anomaly generation. We benchmark multiple training settings with varying levels of anomalous data availability: full supervision with all real anomalies, semi-supervision with few real samples, and settings that rely exclusively on synthetically generated defects produced by a dedicated pipeline. This provides a thorough analysis of the performance-labeling effort trade-off, while allowing us to progressively reduce the reliance on real anomalous data and move closer to the standard VAD paradigm. For each regime, we compare results with state-of-the-art and standard baselines.

(iii) **Dual-branch architecture** for concept-level and pixel-level explanations. VAD approaches provide pixel-level anomaly localization, a capability that standard CBMs lack since they are limited to sample-level predictions. We propose a novel two-branch architecture: a supervised CBM-based Concept branch that delivers concept-level explainability, and a Visual branch that provides pixel-level anomaly localization. This combination pairs the semantic richness of concept-based explanations with the spatial precision of anomaly heatmaps.

(iv) **Cross-dataset robustness evaluation.** We validate our approach across three established industrial VAD benchmarks, MVTec AD, VisA, and Real-IAD, demonstrating the generalizability and robustness of the proposed method across diverse industrial domains and defect types.

The rest of the paper is organized as follows: Sec. 2 reviews relevant VAD literature, while Sec. 3 provides background on the CBM architecture and training methodology. Sec. 4 presents our method, Concept-Aware VAD (CONVAD), detailing its adaptation of CBMs to the VAD setting. Sec. 5 outlines the experimental setting with implementation details and considered supervision scenarios. Sec. 6 discusses the results, and Sec. 7 suggests directions for future work.

2 Related Work

Extensive research in Visual Anomaly Detection (VAD) has led to the development of numerous methods, which can be broadly categorized into four groups:
1. Feature-based Methods: these approaches rely on features extracted from pre-trained neural networks. By leveraging the rich semantic and structural information encoded in these features, anomalies can be detected as deviations from normal patterns. Examples include PatchCore [27], STFPM [37], FastFlow [45] and PaDiM [11].
2. Reconstruction-based Methods: these methods assume models trained only on normal data cannot accurately reconstruct anomalous regions, making reconstruction errors indicative of anomalies. Examples include AnoGAN [29], f-AnoGAN [30], and UniAD [42]. Compared to feature-based methods, they require training generative models, which is computationally expensive.
3. Synthetic Anomaly Methods: these approaches augment datasets with synthetically generated anomalies to learn explicit anomalous patterns. Early methods, such as CutPaste [18], DRAEM [46], and SuperSimpleNet [24], relied on basic or unrealistic augmentations, while recent approaches leverage advanced generative models and Vision Language

Models (VLMs) to produce higher-fidelity anomalies, improving robustness and performance [15, 34]. Notably, the generative model adopted by CONVAD has been validated for realistic industrial anomaly synthesis [15], supporting its suitability for this task.

A less explored setting is the semi-supervised one, with methods such as DevNet [22] and DRA [13], which instead leverage a small number of real anomalous examples alongside normal data, without relying on synthetic generation.

Although these methods provide visual explanations in the form of anomaly maps, they cannot generate semantic human-understandable explanations.

4. VLM-based anomaly detection. Recent VLMs-based approaches combine anomaly detection with natural language description of anomalies [14, 16, 40, 42, 48]. While achieving strong zero-shot performance, which we do not target here, they require VLM inference at the test time, introducing significant computational overhead that may limit industrial deployment. Moreover, zero-shot approaches rely on prior semantic knowledge rather than dataset statistics, potentially classifying semantically unusual but dataset-common patterns as anomalies. For instance, a dataset-wide discoloration may still be flagged as anomalous by a VLM-based method, whereas CONVAD and traditional methods would treat it as normal. In addition, their textual explanations are often post-hoc and may not faithfully reflect the factors driving the prediction, leading to hallucinated or inconsistent explanations [36]. In contrast, CONVAD enforces a semantic bottleneck between visual features and predictions, grounding decisions in explicit interpretable concepts and enabling interventions.

A different direction for improving interpretability is the use of concept-based explanations. Several works explore concept-based methods for detecting and explaining out-of-distribution (OOD) samples in computer vision [10, 19, 31]. However, these methods typically define anomalies at the class level, e.g., treating images of cats as anomalous when the model is trained on dogs. This setting differs from realistic VAD scenarios, where anomalies are usually subtle defects within otherwise normal objects. To the best of our knowledge, no prior work has investigated concept-based methods for VAD. In this work, we address this gap by exploring the use of CBMs for VAD and evaluating their effectiveness.

3 Preliminaries: Concept Bottleneck Models

A CBM [17] is an interpretable neural architecture that makes predictions via an intermediate set of human-interpretable concepts. A key advantage is that CBMs, unlike large VLM, require very limited memory and enable fast inference while still providing human-understandable explanations, making them suitable for resource-constrained or real-time settings. Formally, let $\mathbf{x} \in \mathcal{X}$ be the input, $\mathbf{c} \in \mathcal{C}$ a vector of k concepts, and $y \in \mathcal{Y}$ the target label. A CBM consists of two functions: a concept extractor $g : \mathcal{X} \to \mathcal{C}$, which maps the input $\mathbf{x}$ to a predicted concept vector $\hat{\mathbf{c}} = g(\mathbf{x})$, and a label predictor $f : \mathcal{C} \to \mathcal{Y}$, which maps the predicted concepts $\hat{\mathbf{c}}$ to the final output $\hat{y} = f(\hat{\mathbf{c}})$. The CBM is then the composition $\hat{y} = f(g(\mathbf{x}))$.

The concept layer acts as a bottleneck, forcing predictions $\hat{y}$ to depend on $\mathbf{x}$ entirely through the concept predictions. This improves the interpretability of the downstream output and enables test-time intervention, as discussed below.

Standard CBMs require supervised learning, i.e., require access to ground-truth triplets $\{\mathbf{x}^i, \mathbf{c}^i, y^i\}_{i=1}^m$ during training. In our setting, the input $\mathbf{x}$ is a raw image, $y \in \{0,1\}$ indicates whether an anomaly is present, and true concepts are binary, i.e., $\mathcal{C} = \{0,1\}^k$.

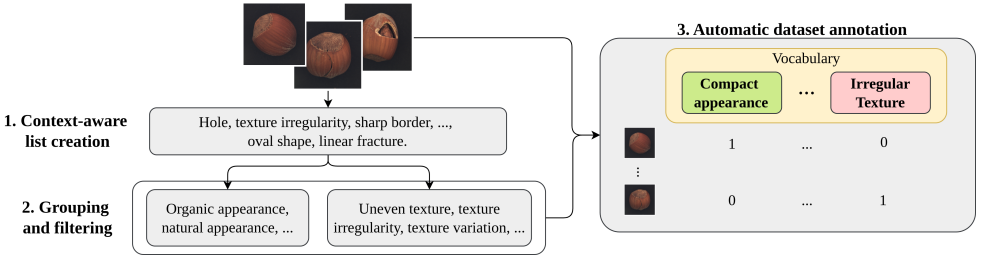


Figure 2: Pipeline for creating the Concept Dataset through concept annotation of a VLM.

3.1 CBM Training

A CBM can be trained using different strategies. In this work, we adopt joint end-to-end training, which consistently outperformed alternative paradigms (independent and sequential training). Details and ablation studies for the alternative paradigms are provided in the Supplementary Material.

In **Joint training**, the concept extractor g and label predictor f are optimized jointly by minimizing

$$\hat{f},\hat{g} = \arg\min_{f,g} \sum_i \mathcal{L}_Y(f(g(\mathbf{x}^i)),y^i) + \lambda \sum_{i,j} \mathcal{L}_{C_j}(g_j(\mathbf{x}^i),c_j^i), \tag{1}$$

where $\mathcal{L}_Y$ is the label prediction loss, $\mathcal{L}_{C_j}$ is the loss for the j-th concept, and λ balances the two objectives.

Note that the predicted concepts $g_j(\mathbf{x}^i)$ are continuous logits rather than binary variables as in the ground-truth concept vectors. The classifier f operates directly on these logits.

3.2 CBM Test-time Interventions

A key property of CBMs is their support for test-time interventions. When a user or domain expert identifies incorrectly predicted concepts, they can manually correct them by setting $\hat{c}_j = c_j$. The resulting correct concept vector $\tilde{\mathbf{c}}$ is fed into the label predictor f, obtaining an updated (and ideally more accurate) output $\tilde{y} = f(\tilde{\mathbf{c}})$. In jointly trained models, where f operates on concept logits, interventions replace the corresponding logits with the 5th or 95th percentile of the training logit distribution to avoid extreme values (see Supplementary Material for more details). This mechanism allows experts to improve predictions without retraining the model, a capability enabled by the CBMs bottleneck structure.

4 Methodology

4.1 Concept Dataset Pipeline

A key challenge of CBMs is the lack of concept-annotated datasets, which are unavailable in many domains, including VAD. Prior works addressed this using automated pipelines based on VLMs, such as Label-free CBMs [21], LaBo [43], and VLG-CBMs [53]. However, our setting differs from prior work, which focused on multiclass classification of natural images. In VAD, concepts are domain-specialized and fine-grained. Moreover, industrial anomalies are highly dataset- and category-dependent and are underrepresented in VLM training

data, which can lead to generic or hallucinated concepts. We therefore devise a dedicated anomaly-oriented concept extraction pipeline and prompt (see details in the Supplementary Material), and we validate it against a manually annotated ground truth. Our approach draws inspiration from existing concept-extraction methods in other domains, with minimal adaptations for VAD.

Our procedure, shown in Fig. 2, consists of three steps:

1. Context-aware list creation. Unlike existing approaches, which generate concepts without domain context, we introduce a context-aware paradigm tailored to each object category. We sample a representative subset $X = \{x_i \in D \mid i = 1,\dots,N\}$ (5% of the dataset) ensuring coverage of all defect types. Let $\mathcal{M}$ be the VLM and pt a textual prompt that injects contextual information (object category, defect type) and employs a Chain-of-Thought strategy [39] (see Supplementary Material for details on the prompt design). The model first generates a natural language image description and then extracts up to five semantic concepts from it. By constraining extraction to a visual description, we prevent the model from focusing on non-visual attributes and limit hallucinations. The resulting concept set is:

$$\mathcal{V} = \{v_j^i \mid i = 1,\dots,N;\ j = 1,\dots,|v^i|\}, \qquad v^i = \mathcal{M}(x^i, pt). \tag{2}$$

2. Grouping and filtering. The raw concepts are refined in two substeps: (i) **Concept grouping**: the full list is fed back to the VLM with few-shot examples [7] to merge morphologically and semantically related attributes, reducing the set from a few hundred to fewer than fifty concepts. (ii) **Concept filtering**: following [21], we encode concepts with the CLIP ViT-B/32 text encoder E_T [23] and remove near-duplicates by discarding one concept from each pair exceeding a cosine similarity of 0.9:

$$\cos(E_T(v_i), E_T(v_j)) = \frac{E_T(v_i) \cdot E_T(v_j)}{\|E_T(v_i)\|\ \|E_T(v_j)\|} \tag{3}$$

3. Automatic dataset annotation. Each image is annotated with binary concept labels. A VLM is queried with a category- and defect-aware prompt to inspect each image against the final concept list and return a boolean assignment per concept (see Supplementary Material). The quality of the automatic annotation process is tested against a manually annotated subset of 400 images across all categories, achieving 91.3% accuracy (see Supplementary Material for further details).

Fig. 3 shows the resulting vocabulary and its distribution for the Hazelnut category in MVTec, divided by defect types. Concepts exhibit meaningful differences between the normal and anomalous class, confirming their informativeness for the downstream task. Anomalous-class concepts include both general indicators (*uneven tone, surface discontinuity*) and fine-grained descriptors (*visible ink, dark interior, hole*), with the latter exhibiting distinct distributions across different defect types and therefore enabling more precise and discriminative characterization of each specific defect. Notably, concepts are not exclusive to normal or abnormal classes; many appear in both with different frequencies. Thus, there is no one-to-one mapping between individual concepts and the final label, and classification relies on their joint interactions rather than individual concepts.

4.2 CONVAD architecture

Modern VAD approaches provide pixel-level anomaly localization, whereas standard CBMs cannot. We bridge this gap by augmenting CBMs with a spatial localization branch, enabling

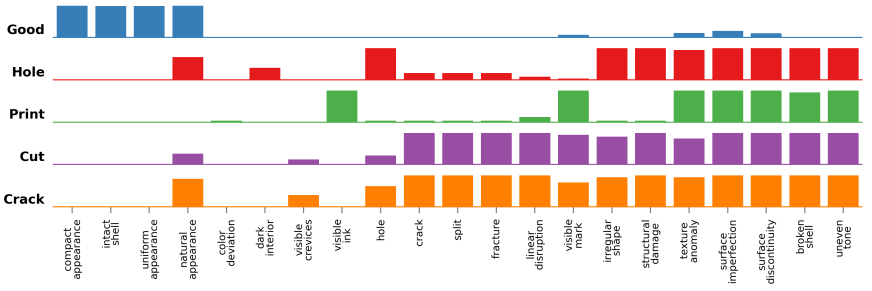


Figure 3: Example of concept vocabulary and distribution across defect types for the MVTec hazelnut category.

them to jointly produce concept-based explanations and fine-grained anomaly heatmaps. The proposed architecture, illustrated in Fig. 1, consists of two branches: (i) a *Concept branch* ϕ_{cbm}, which produces a concept prediction and an anomaly score, and (ii) a *Visual branch* ϕ_{vis}, trained as a student to match the Concept-branch features on normal data and producing spatial anomaly localization together with an additional anomaly score. The two anomaly scores are merged as in Eq. 9. Sharing the feature extractor across branches encourages consistency between semantic and spatial explanations.

4.2.1 Training

Training proceeds in two sequential phases. First, ϕ_{cbm} is trained end-to-end on concept prediction and anomaly detection objectives. Then, its weights are frozen and ϕ_{vis} is trained as a student to match the intermediate features of ϕ_{cbm} on normal samples only, following the student–teacher VAD paradigm [3, 12, 37, 47].

Concept Branch. We define per-concept and anomaly detection losses:

$$\mathcal{L}_{C_j} = \mathcal{L}_{\text{BCE}}(c_j, \hat{c}_j, \alpha), \qquad \mathcal{L}_Y = \mathcal{L}_{\text{BCE}}(y, \hat{y}, \alpha). \tag{4}$$

where $\mathcal{L}_{\text{BCE}}(z, \hat{z}; \alpha) = -\frac{1}{n}\sum_{i=1}^{n}(\alpha z^i \log \hat{z}^i + (1-z^i)\log(1-\hat{z}^i))$.
The weighting factor $\alpha \in \mathbb{R}$ compensates for class imbalance in both concept and anomaly predictions, and is computed as the imbalance ratio $\alpha = (n - \sum_{i=1}^{n} z^i)/(\sum_{i=1}^{n} z^i)$. The two losses are combined as:

$$\mathcal{L}_{cbm} = \frac{1}{1+\lambda k}\left(\mathcal{L}_Y + \lambda \sum_{j=1}^{k} \mathcal{L}_{C_j}\right) \tag{5}$$

where k denotes the number of concepts and $\lambda \in \mathbb{R}$ controls the trade-off between the anomaly detection and concept prediction objectives.

Visual Branch. Once ϕ_{cbm} is frozen, ϕ_{vis} is optimized to match its intermediate feature representations on normal samples only:

$$\mathcal{L}_{\text{vis}} = \frac{1}{|\mathcal{X}_n|}\sum_{\mathbf{x}\in\mathcal{X}_n}\frac{1}{F}\sum_{f=1}^{F}\left\|\phi_{\text{cbm}}^{(f)}(\mathbf{x}) - \phi_{\text{vis}}^{(f)}(\mathbf{x})\right\|_2^2, \tag{6}$$

where $\mathcal{X}_n$ denotes the set of normal training samples, F is the fixed number of intermediate feature layers [3, 37], and $\phi_{\text{cbm}}^{(f)}, \phi_{\text{vis}}^{(f)}$ indicate the f-th layer of the respective network.

4.2.2 Inference

Given a test image $\mathbf{x}$, the model produces three outputs: a concept-based explanation, a spatial heatmap and an anomaly score.

Concept-based Explanation. Given the predicted concept activations $\hat{\mathbf{c}} = (\hat{c}_1, \ldots, \hat{c}_k)$ and the learned weights $\mathbf{w} = (w_1, \ldots, w_k)$ of the final linear layer, the contribution of each concept c_j to the anomaly score is proportional to:

$$\delta_j(\mathbf{x}) = w_j \cdot \hat{c}_j(\mathbf{x}). \tag{7}$$

The explanation is then presented as the ranked list of concepts sorted by $|\delta_j(\mathbf{x})|$, highlighting those that most influenced the final prediction.

Compared to classic VAD approaches, CONVAD has two key advantages. First, the model is inherently interpretable, as ranked concept contributions provide a direct, human-readable justification for the anomaly prediction. Second, the most relevant concept activations $\hat{\mathbf{c}}$ can be exposed to human experts for validation, enabling human-in-the-loop intervention by correcting mispredicted concepts and directly influencing the final decision without retraining [9].

Anomaly Heatmap. At inference, both networks process the input image $\mathbf{x}$, and an anomaly heatmap is computed from the discrepancy between their feature maps, as anomalous regions lie outside the student training distribution:

$$H_{\text{vis}}(\mathbf{x}) = \frac{1}{F}\sum_{f=1}^{F} \mathcal{U}\left(D^{(f)}(\mathbf{x})\right), \qquad D^{(f)}_{\text{h,w}}(\mathbf{x}) = \left\| \phi^{(f)}_{\text{cbm,h,w}}(\mathbf{x}) - \phi^{(f)}_{\text{vis,h,w}}(\mathbf{x}) \right\|_2^2, \tag{8}$$

where $\mathcal{U}(\cdot)$ denotes bilinear upsampling to the original resolution $H \times W$ and the per-layer squared difference is averaged across feature channels. This approach pairs the textual explanations produced by the CBM branch with a spatial visualization of anomalous regions. A schematic illustration is shown in Fig. 1.

Anomaly Score. To align the two branches onto a common scale, we calibrate the Visual branch using a small held-out set $\mathcal{X}_{\text{cal}}$. The per-image visual score is obtained by max-pooling the heatmap over spatial dimensions, $s_{\text{vis}}(\mathbf{x}) = \max_{h,w} \mathbf{H}^{(h,w)}_{\text{vis}}(\mathbf{x})$, and the threshold τ^* maximising F1 on $\mathcal{X}_{\text{cal}}$ yields the binarized prediction $\hat{y}_{\text{vis}}(\mathbf{x}) = \mathbf{1}[s_{\text{vis}}(\mathbf{x}) \geq \tau^*] \in \{0,1\}$. The Concept branch outputs a sigmoid anomaly probability $s_{\text{cbm}}(\mathbf{x}) = \sigma(\phi_{\text{cbm}}(\mathbf{x})) \in [0,1]$ conditioned on $\hat{\mathbf{c}}(\mathbf{x})$. The final score combines both signals:

$$s(\mathbf{x}) = \max\left(\hat{y}_{\text{vis}}(\mathbf{x}),\ s_{\text{cbm}}(\mathbf{x})\right). \tag{9}$$

An anomaly detected by either branch is sufficient to flag the image, making detection more resilient to failures of a single branch. Notably, as shown in the right subtable of Table 2, the Concept branch drives the final prediction in approximately 88–91% of cases across supervision levels, meaning the Visual branch acts primarily as a fallback for out-of-distribution samples not captured by the concept layer. As a result, the vast majority of detections are driven by concept-level explanations. An ablation regarding alternative fusion strategies is available in the Supplementary Material, displaying that, while achieving a higher AUROC, they consistently reduce agreement with the concept branch.

4.3 Synthetic Anomaly Generation

In line with prior works [15, 34], we adopt a generative approach for realistic synthetic anomaly generation. By incorporating these synthetic samples, CONVAD can operate across all supervision regimes, including semi-supervised and unsupervised settings where no real anomalous images are available.
In detail, synthetic anomalous images are created by modifying normal images using a text-to-image and image-editing system. Let $\mathcal{G}_{\text{edit}}(\mathbf{x}, p)$ denote the editing function, which takes $\mathbf{x} \in \mathbb{R}^{H\times W\times 3}$ as input image and p that represents a textual prompt. Starting from a normal image $\mathbf{x}_n$, the generative procedure creates an anomalous image $\mathbf{x}_a = \mathcal{G}_{\text{edit}}(\mathbf{x}_n, p_a)$ where p_a includes the desired defect type and ensures consistency with the original image. The scope of this operation is to introduce anomalies while preserving the rest of the image unchanged.

Prompts are carefully designed to control the generation of anomalies (see Supplementary Material for further details). Each prompt specifies: (i) Defect type: the type of anomaly to introduce (e.g., scratch, crack, dent). (ii) Object type: the object type or surface being modified (e.g., metal plate, plastic part). (iii) Pose and view details: information that ensures the edited image maintains the same camera angle, orientation, and lighting as the original.

5 Experimental Setting

5.1 Scenarios and baselines

In this work, we evaluate CONVAD under several scenarios that differ in the type and amount of supervision provided during training. We adopt the same definitions as [41].
Supervised: training on both normal and anomalous images from the real-world dataset. We compare against supervised anomaly detection (AD) based on UNet [26], due to the absence of specific fully-supervised VAD methods that train on the same datasets and training conditions.
Semi-Supervised: training on all available normal images and synthetically generated anomalies, together with a limited number of real anomalous samples. Specifically, we adopt a one-shot setting per defect type, resulting in an average of 5 real anomalous images per category. For comparison, we evaluate against the semi-supervised approaches DevNet [22] and DRA [13] under the same setting, using 5 anomalous images per category.
Unsupervised: training uses only normal images from the real-world dataset, together with synthetically generated samples. Following established prior works [18, 20, 41], this setting is still considered unsupervised since no real anomalous samples are used during training. In this scenario, concepts are extracted exclusively from synthetically generated anomalous images, so the model never accesses real defective samples during concept extraction; moreover, calibration of the F1-maximizing threshold for score fusion uses only synthetic anomalies and normal images, while inference is performed on real anomalous images, keeping calibration and test data fully disjoint. We compare against PatchCore [27], STFPM [32], Tiny-Dinomaly [5], AnomalyGPT [14] and the zero-shot method Anomaly-OV [42]. Note that Anomaly-OV operates in a fully zero-shot manner, requiring no training images whatsoever, whereas AnomalyGPT, despite being VLM-based, requires normal reference images at inference time and therefore falls under the unsupervised scenario.

5.2 Implementation Details

Models. In the Concept Dataset Annotation Pipeline, we mainly leverage Gemma3 [35] as a VLM, except in the Concept Filtering phase, where we use the CLIP ViT-B/32 text encoder [23]. For the text-to-image and image-editing system in the synthetic anomaly generation pipeline, we employ Gemini 2.5 Flash Image (Nano Banana). The CBM uses MobileNet-v2 as a feature extractor [28].
PatchCore and STFPM are evaluated using the same MobileNet backbone as our approach, similarly to the UNet, which was built using MobileNet-v2 as an encoder to ensure comparability. Tiny-Dinomaly adopts a lightweight DeiT backbone, while AnomalyGPT uses Vicuna-7B with ImageBind-Huge as image encoder, and Anomaly-OV uses LLaVA-One-Vision 0.5B, as in the original papers.
Datasets. Our main experimental results are obtained using the MVTec AD dataset, and further results on industrial benchmarks, including VisA and Real-IAD, confirm cross-dataset robustness.
Hyperparameter Optimization. To select the best hyperparameter configuration, we employ Bayesian optimization, implemented through the Optuna library [1], using the Tree-structured Parzen Estimator (TPE) algorithm [33] (details in Supplementary Material).
Training Details. CONVAD results are averaged over three runs with different random seeds. Inference uses a held-out set composed of 20% of the real-world dataset. Additional details about training and data augmentation in Supplementary Material. The full implementation is available in the open-source library MoViAD [4].

5.3 Metrics

VAD algorithms can be assessed using multiple criteria, corresponding to different levels of analysis:
Image-level: To evaluate detection at the image level (each image is a different sample), we report the AUROC (Area under the Receiver Operating Characteristic curve) score, referred to as I-AUC.
Pixel-level: To assess performance at the pixel level, we consider the commonly used AUROC score, named P-AUC, computed by considering each pixel of each image as an independent sample.
Concept-level: For CONVAD, we evaluate the accuracy of concept predictions using the average AUROC score, referred to as C-AUC. Since concepts are binary, we compute the AUROC score for each concept separately and then combine them by averaging over all concepts. Each term is weighted by w_j, i.e., the number of true instances for the $j-$th concept c_j, where k is the total number of concepts:

$$\text{C-AUC} := \frac{\sum_{j=1}^{k} w_j \cdot \text{AUROC}(c_j)}{\sum_{j=1}^{k} w_j}, \tag{10}$$

6 Results

Sec. 6.1 compares CONVAD with state-of-the-art VAD approaches across supervision regimes. Sec. 6.2 reports a user study assessing perceived explanation quality. Sec. 6.3 evaluates concept-level interventions as a mechanism for human-machine collaboration. Finally, Sec. 6.4 ablates the contribution of the Concept and Visual branches and analyzes the pixel-level performance of CONVAD.

Table 1: Comparison of CONVAD and other VAD methods across unsupervised, semi-supervised, and supervised scenarios on the MVTec AD dataset. ✗ indicates that the method cannot produce the corresponding output (pixel localization or concept-level prediction).

Metric	Zero-Shot	Unsupervised (Zero-Anomaly-Shot)					Semi-Supervised			Supervised	
	Anomaly-OV	CONVAD	PatchCore	STFPM	Tiny-Dinomaly	AnomalyGPT	CONVAD	DevNet	DRA	CONVAD	UNet
I-AUC	0.94	0.81	0.86	0.88	0.91	**0.97**	**0.87**	0.80	0.76	**0.96**	0.65
P-AUC	✗	0.91	**0.96**	0.88	**0.96**	0.93	**0.92**	**0.92**	✗	**0.95**	0.77
C-AUC	✗	**0.73**	✗	✗	✗	✗	**0.72**	✗	✗	**0.86**	✗
Memory [MB] ↓	1,768	17.6	38.4	**11.1**	36.5	30,520	**17.6**	44.7	44.7	17.6	**10.3**

6.1 CBM Scenarios

A comparison of CONVAD against state-of-the-art VAD approaches on MVTec AD is presented in Table 1 across supervised, semi-supervised, and unsupervised scenarios; additional results on VisA and Real-IAD datasets are reported in Table 2 (Left).

Supervised. When compared with the traditional UNet baseline in the supervised setting, CONVAD achieves improved performance while additionally providing concept-based explanations that make predictions interpretable to domain experts. Although the supervised setting has access to all available anomalous samples, these remain scarce, thereby limiting the performance of UNet and highlighting the data efficiency of our method. Further results in the supervised setting across the three considered benchmarks (MVTec AD, VisA, Real-IAD) confirm the effectiveness of CONVAD.
CONVAD and UNet have a comparable memory footprint (17.6 MB vs. 10.3 MB); however, UNet achieves substantially lower detection performance (I-AUC: 0.65 vs. 0.96, P-AUC: 0.77 vs. 0.95) despite requiring full supervision, suggesting that its limited generalization makes it an impractical choice for real-world industrial inspection.
Semi-supervised. CONVAD outperforms DevNet and DRA, two established semi-supervised VAD baselines. This suggests that the concept-based inductive bias embedded in CONVAD acts as an effective regularizer, improving resilience to scarce anomalous training data and generalization under limited supervision. A key enabler is the synthetic anomaly generation procedure, which produces samples that closely align with the real abnormal domain (Fig. 5), providing the model with more representative concept supervision despite limited real anomalous data. See Sec. 6.5 for details.
Unsupervised. CONVAD reaches an I-AUC of 0.81, trailing the similarly-sized STFPM (0.88) and Tiny-Dinomaly (0.91), the latter requiring twice the memory (36.5 MB vs. 17.6 MB). We consider this an acceptable cost, as none of these baselines exposes concept-level explanations and the gap is largely recovered through interventions (Sec. 6.3). Methods with higher detection scores come with significant trade-offs: AnomalyGPT achieves the best I-AUC (0.97) but requires over 30 GB of memory (nearly 1,750× more than CONVAD) and ranks as the worst method in our user study on explanation quality (Sec. 6.2), with 44.2% of its explanations rated as unhelpful. Similarly, Anomaly-OV attains strong detection performance (I-AUC: 0.94) but produces no pixel-level heatmaps, which are a fundamental output in industrial VAD applications. In contrast, CONVAD delivers all three outputs: concept explanations, pixel heatmaps, and anomaly scores with a compact footprint, and benefits from human-in-the-loop interventions that can boost detection performance at inference time (Sec. 6.3).

Table 2: (Left) Average results on MVTec AD, VisA, and Real-IAD (Supervised) for the Concept branch vs. PatchCore (WideResNet-50, although unsupervised, is a strong baseline) and AnomalyGPT (4-normal-shot). (Right) Ablation of CONVAD on MVTec AD across supervision levels; I-AUC reported for the Concept branch, Visual branch, and their fusion. Concept Rate is the fraction of samples where the Concept branch drives the final prediction.

	CONVAD		PatchCore	AnomalyGPT
Dataset	**C-AUC**	**I-AUC**	**I-AUC**	**I-AUC**
MVTec AD	0.86	0.97	**0.99**	0.96
VisA	0.76	**0.94**	0.93	0.91
Real-IAD	0.88	**0.94**	0.90	—

Sup. Level	**I-AUC**			**Concept**
	Concept	**Visual**	**Fusion**	**Rate (%)**
Unsuper.	0.77	0.90	0.81	88.3
Semi-Sup.	0.85	0.91	0.87	90.1
Supervised	0.97	0.96	0.96	91.3

Table 3: User study results (582 responses, 32 participants). **Left**: perceived helpfulness of explanations. **Right**: explanation quality signals; *effective*: the explanation enabled the user to verify whether the prediction was correct; *misleading*: it convinced the user to trust a wrong prediction (counting defined in Eq. 11).

Method	Helpful	Partial	Unhelpful ↓
AnomalyGPT [14]	33.2%	22.6%	44.2%
Anomaly-OV [42]	**59.2%**	22.5%	18.3%
CONVAD	55.6%	28.6%	**15.8%**

Signal	AGPT	A-OV	CONVAD
Effective	33.2%	60.7%	**73.5%**
Misled ↓	18.4%	11.7%	**3.1%**
Useless ↓	48.4%	27.6%	**23.5%**

6.2 User Study

We conducted an online study with 32 participants to assess perceived explanation quality on MVTec and VisA images (See Supplementary Material for more details). Each participant was shown a product image with the model prediction, its explanation and, when available, the anomaly heatmap, then asked (Q1) if the prediction appeared correct and (Q2) if the explanation was helpful. Responses were collected for CONVAD, AnomalyGPT [14], and Anomaly-OV [42], balanced across methods and categories (582 responses total).

We combine both answers into a single quality signal. For a response, let $q_1 \in \{\text{yes}, \text{no}, \text{unsure}\}$ encode the user's verdict on the prediction (Q1), $q_2 \in \{\text{yes}, \text{partial}, \text{no}\}$ the perceived helpfulness (Q2), and $m \in \{0,1\}$ whether the prediction actually matched the ground truth (logged server-side, hidden from the user). An explanation is *effective* when it is helpful and the user's verdict agrees with the truth, and *misleading* when it is helpful but steers the user against the truth. Over N responses,

$$\begin{aligned} \text{Effective} &= \tfrac{1}{N}\big|\{\, q_1{\neq}\text{unsure} \wedge q_2{\neq}\text{no} \wedge (q_1{=}\text{yes} \Leftrightarrow m{=}1)\,\}\big| \\ \text{Misled} &= \tfrac{1}{N}\big|\{\, q_1{\neq}\text{unsure} \wedge q_2{\neq}\text{no} \wedge (q_1{=}\text{yes} \Leftrightarrow m{=}0)\,\}\big|, \end{aligned} \tag{11}$$

and the remainder is *useless* ($q_2{=}$no or $q_1{=}$unsure), so Useless $= 1 -$ Effective $-$ Misled.

In safety-critical industrial inspection, a useful explanation must enable the operator to independently verify whether the prediction is correct, catching both false alarms and missed defects before they propagate downstream.

As reported in Table 3, CONVAD rates as helpful or partially helpful in 84.2% of cases, on par with Anomaly-OV (81.7%) despite a slightly lower outright helpfulness rate (55.6% vs. 59.2%). More importantly, CONVAD achieves the lowest unhelpful rate (15.8% vs. 18.3% and 44.2%) and the fewest misleading explanations (3.1% vs. 11.7% and 18.4%), indicating that Anomaly-OV's marginal helpfulness advantage comes at the cost of substantially greater risk of misleading operators. The intrinsic coupling between concept scores and the anomaly prediction, as opposed to post-hoc verbalization in VLM-based methods, reduces

noise in the explanatory signal and prevents errors from being masked by plausible-sounding text.

6.3 Intervention

We evaluate concept-level interventions in the **unsupervised** setting, where the model has the least prior knowledge of the anomalous domain and thus stands to benefit most from human feedback. As shown in Fig. 4, anomaly detection performance improves consistently as more concepts are intervened upon, with most MVTec categories approaching near-perfect AUC once a majority of concepts are corrected; this confirms the value of human-in-the-loop refinement for practical and safety-critical applications, where practitioners can trade off annotation effort against detection accuracy.

Two categories, *transistor* and *toothbrush*, show a degradation: both have few concepts, making the hard label imposed by intervention over-constrain the prediction. Nevertheless, the overall trend is positive, and this capability is a unique advantage of CONVAD over standard VAD methods that, lacking a concept bottleneck, cannot expose any mechanism for structured human correction.

6.4 Ablation Study

Table 2 (right) presents an ablation of CONVAD across supervision levels, reporting the contribution of the Concept branch, the Visual branch, and their fusion. Our Visual branch improves over STFPM at the pixel level: as shown in Table 1, P-AUC increases from 0.88 (STFPM, MobileNet-v2) to 0.91, thanks to teacher network fine-tuning on the concept prediction task, which injects domain knowledge and yields more informative feature extraction [3]. With respect to PatchCore, our Visual branch trades a lower P-AUC (0.91 vs. 0.96) for a higher I-AUC (0.90 vs. 0.86) at less than half the memory, and adds concept-based explanations. Remarkably, the Visual branch not only provides intuitive visual explanations to the end user but also enhances the robustness of the model when the Concept branch fails. Moreover, because the Visual branch of CONVAD is based on the student–teacher paradigm, which has been proven effective for detecting novel data, it further improves robustness against previously unseen anomalies that the Concept branch might miss. Importantly, CONVAD is not tied to any specific Visual branch architecture: any method that operates on a shared feature extractor can serve as the Visual branch, making the methodology broadly applicable. STFPM was chosen here as a representative example due to its efficiency.

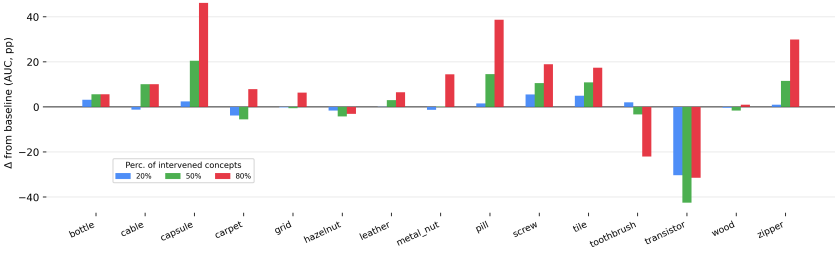


Figure 4: Performance gain in MVTec categories for increasing number of intervened concepts. Unsupervised setting.

6.5 Synthetic-to-real Alignment

The usefulness of generated anomalies hinges on whether their concept-logit representations align with those of real defects. We probe this with 2D t-SNE embeddings of the concept logits on `metal_nut`, a visually complex category (Fig. 5). In the unsupervised setting, synthetic anomalies form a tight cluster separated from normal samples, while real test anomalies fall largely *within* the normal region: generated and real defects are misaligned, and concepts transfer poorly (C-AUC 0.59, I-AUC 0.66). This is specific to hard categories: in simpler ones such as hazelnut, synthetic and real anomalies overlap and transfer well (see Supplementary Material). Adding a single real anomaly per defect type (semi-supervised) pulls real anomalies toward the synthetic cluster, raising performance to C-AUC 0.70 and I-AUC 0.84: even minimal supervision is highly effective at bridging the synthetic-to-real gap.

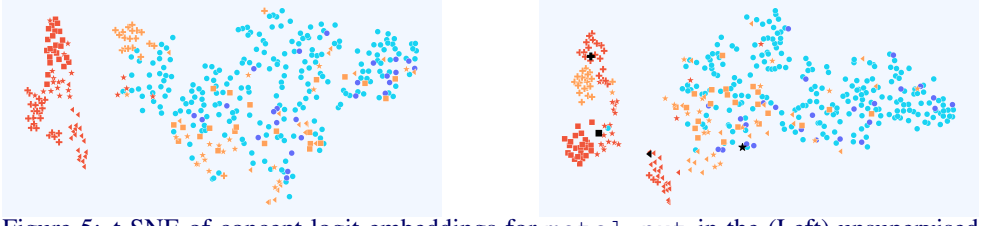


Figure 5: t-SNE of concept-logit embeddings for `metal_nut` in the (Left) unsupervised and (Right) semi-supervised settings. Color encodes the split/label (● train normal, ● train anomalous (generated), ● test normal, ● test anomalous (real), ● train anomalous (real)), while marker shape encodes the defect type (● good, ■ bent, ◀ color, + flip, ★ scratch). Generated anomalies (Left) cluster apart from the real test anomalies; adding one real anomaly per defect type (Right) pulls real and synthetic anomalies into alignment.

7 Conclusion

This work introduces CONVAD, adapting CBMs to VAD to bridge high-performing detection and human-understandable reasoning. Pairing a supervised CBM-based Concept branch for concept-level explanations with a Visual branch for pixel-level localization, CONVAD delivers dual interpretability that no existing VAD method offers.

Across MVTec AD, VisA, and Real-IAD, CONVAD matches state-of-the-art detection while providing transparent, concept-grounded predictions. Concept-level interventions yield substantial gains, confirming the value of the human-machine collaboration the CBM formulation enables. Benchmarking supervision regimes shows even an unsupervised fully synthetic-anomalies setting is viable, while combining few real anomalies with synthetic defects is a robust, cost-effective strategy.

The main limitation is the reliance on anomalous training samples: synthetic generation reduces this dependency but does not yet match real anomalies. Future work will improve synthetic generation quality and exploit the Visual branch for novelty detection, capturing unseen defects where the Concept branch alone fails, closing the gap toward fully unsupervised, interpretable and efficient VAD. Moreover, we plan on devising a strategy to effectively unify the concept and visual branch so that they process the same signal, without incurring a loss in performance.

References

[1] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 2623–2631, 2019.

[2] Jinan Bao, Hanshi Sun, Hanqiu Deng, Yinsheng He, Zhaoxiang Zhang, and Xingyu Li. Bmad: Benchmarks for medical anomaly detection, 2024. URL https://arxiv.org/abs/2306.11876.

[3] Manuel Barusco, Francesco Borsatti, Davide Dalle Pezze, Francesco Paissan, Elisabetta Farella, and Gian Antonio Susto. Paste: Improving the efficiency of visual anomaly detection at the edge. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 4026–4035, 2025.

[4] Manuel Barusco, Francesco Borsatti, Arianna Stropeni, Davide Dalle Pezze, and Gian Antonio Susto. Moviad: A modular library for visual anomaly detection. *arXiv preprint arXiv:2507.12049*, 2025.

[5] Manuel Barusco, Francesco Borsatti, David Petrovic, Davide Dalle Pezze, and Gian Antonio Susto. Continual visual anomaly detection on the edge: Benchmark and efficient solutions. *arXiv preprint arXiv:2604.06435*, 2026.

[6] 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.

[7] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[8] Robin Chan, Krzysztof Lis, Svenja Uhlemeyer, Hermann Blum, Sina Honari, Roland Siegwart, Pascal Fua, Mathieu Salzmann, and Matthias Rottmann. Segmentmeifyoucan: A benchmark for anomaly segmentation, 2021. URL https://arxiv.org/abs/2104.14812.

[9] Kushal Chauhan, Rishabh Tiwari, Jan Freyberg, Pradeep Shenoy, and Krishnamurthy Dvijotham. Interactive concept bottleneck models. In *Proceedings of the aaai conference on artificial intelligence*, volume 37, pages 5948–5955, 2023.

[10] Jihye Choi, Jayaram Raghuram, Ryan Feng, Jiefeng Chen, Somesh Jha, and Atul Prakash. Concept-based Explanations for Out-of-Distribution Detectors. In *Proceedings of the 40th International Conference on Machine Learning*, pages 5817–5837. PMLR, July 2023. URL https://proceedings.mlr.press/v202/choi23e.html. ISSN: 2640-3498.

[11] Thomas Defard, Aleksandr Setkov, Angelique Loesch, and Romaric Audigier. Padim: a patch distribution modeling framework for anomaly detection and localization, 2020. URL https://arxiv.org/abs/2011.08785.

[12] Hanqiu Deng and Xingyu Li. Anomaly detection via reverse distillation from one-class embedding, 2022. URL https://arxiv.org/abs/2201.10703.

[13] Choubo Ding, Guansong Pang, and Chunhua Shen. Catching both gray and black swans: Open-set supervised anomaly detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7388–7398, 2022.

[14] 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.

[15] Jinwei Hu, Francesco Borsatti, Arianna Stropeni, Davide Dalle Pezze, Manuel Barusco, and Gian Antonio Susto. Mirage: Model-agnostic industrial realistic anomaly generation and evaluation for visual anomaly detection. *arXiv preprint arXiv:2603.13507*, 2026.

[16] 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 (CVPR)*, pages 11375–11385, June 2024.

[17] Pang Wei Koh, Thao Nguyen, Yew Siang Tang, Stephen Mussmann, Emma Pierson, Been Kim, and Percy Liang. Concept bottleneck models. In *International conference on machine learning*, pages 5338–5348. PMLR, 2020.

[18] Chun-Liang Li, Kihyuk Sohn, Jinsung Yoon, and Tomas Pfister. Cutpaste: Self-supervised learning for anomaly detection and localization, 2021. URL https://arxiv.org/abs/2104.04015.

[19] Tongkun Liu, Bing Li, Xiao Du, Bingke Jiang, Xiao Jin, Liuyi Jin, and Zhuo Zhao. Component-aware anomaly detection framework for adjustable and logical industrial visual inspection. *Advanced Engineering Informatics*, 58:102161, October 2023. ISSN 1474-0346. doi: 10.1016/j.aei.2023.102161. URL https://www.sciencedirect.com/science/article/pii/S1474034623002896.

[20] 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.

[21] Tuomas Oikarinen, Subhro Das, Lam M Nguyen, and Tsui-Wei Weng. Label-free concept bottleneck models. *arXiv preprint arXiv:2304.06129*, 2023.

[22] Guansong Pang, Choubo Ding, Chunhua Shen, and Anton van den Hengel. Explainable deep few-shot anomaly detection with deviation networks. *arXiv preprint arXiv:2108.00462*, 2021.

[23] 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.

[24] Blaž Rolih, Matic Fučka, and Danijel Skočaj. Supersimplenet: Unifying unsupervised and supervised learning for fast and reliable surface defect detection. In *International Conference on Pattern Recognition*, pages 47–65. Springer, 2024.

[25] Blaž Rolih, Matic Fučka, and Danijel Skočaj. No label left behind: a unified surface defect detection model for all supervision regimes. *Journal of Intelligent Manufacturing*, pages 1–21, 2025.

[26] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[27] Karsten Roth, Latha Pemula, Joaquin Zepeda, Bernhard Schölkopf, Thomas Brox, and Peter Gehler. Towards total recall in industrial anomaly detection, 2022. URL https://arxiv.org/abs/2106.08265.

[28] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018.

[29] Thomas Schlegl, Philipp Seeböck, Sebastian M. Waldstein, Ursula Schmidt-Erfurth, and Georg Langs. Unsupervised anomaly detection with generative adversarial networks to guide marker discovery, 2017. URL https://arxiv.org/abs/1703.05921.

[30] Thomas Schlegl, Philipp Seeböck, Sebastian M. Waldstein, Georg Langs, and Ursula Schmidt-Erfurth. f-anogan: Fast unsupervised anomaly detection with generative adversarial networks. *Medical Image Analysis*, 54:30–44, 2019. ISSN 1361-8415. doi: https://doi.org/10.1016/j.media.2019.01.010. URL https://www.sciencedirect.com/science/article/pii/S1361841518302640.

[31] Laya Rafiee Sevyeri, Ivaxi Sheth, Farhood Farahnak, Samira Ebrahimi Kahou, and Shirin Abbasinejad Enger. Transparent Anomaly Detection via Concept-based Explanations, November 2023. URL http://arxiv.org/abs/2310.10702. arXiv:2310.10702 [cs].

[32] Sungbin Shin, Yohan Jo, Sungsoo Ahn, and Namhoon Lee. A closer look at the intervention procedure of concept bottleneck models. In *International Conference on Machine Learning*. PMLR, 2023.

[33] Divyansh Srivastava, Ge Yan, and Lily Weng. Vlg-cbm: Training concept bottleneck models with vision-language guidance. *Advances in Neural Information Processing Systems*, 37:79057–79094, 2024.

[34] Han Sun, Yunkang Cao, Hao Dong, and Olga Fink. Unseen visual anomaly generation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 25508–25517, 2025.

[35] Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, et al. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*, 2025.

[36] Miles Turpin, Julian Michael, Ethan Perez, and Samuel Bowman. Language models don't always say what they think: Unfaithful explanations in chain-of-thought prompting. *Advances in Neural Information Processing Systems*, 36:74952–74965, 2023.

[37] Guodong Wang, Shumin Han, Errui Ding, and Di Huang. Student-teacher feature pyramid matching for anomaly detection, 2021. URL https://arxiv.org/abs/2103.04257.

[38] Shuhei Watanabe. Tree-structured parzen estimator: Understanding its algorithm components and their roles for better empirical performance. *arXiv preprint arXiv:2304.11127*, 2023.

[39] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed H. Chi, F. Xia, Quoc Le, and Denny Zhou. Chain of thought prompting elicits reasoning in large language models. *ArXiv*, abs/2201.11903, 2022. URL https://api.semanticscholar.org/CorpusID:246411621.

[40] Peng Wu, Xuerong Zhou, Guansong Pang, Lingru Zhou, Qingsen Yan, Peng Wang, and Yanning Zhang. Vadclip: Adapting vision-language models for weakly supervised video anomaly detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 6074–6082, 2024.

[41] 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*, 54(5):2720–2733, 2024.

[42] Jiacong Xu, Shao-Yuan Lo, Bardia Safaei, Vishal M Patel, and Isht Dwivedi. Towards zero-shot anomaly detection and reasoning with multimodal large language models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 20370–20382, 2025.

[43] Yue Yang, Artemis Panagopoulou, Shenghao Zhou, Daniel Jin, Chris Callison-Burch, and Mark Yatskar. Language in a bottle: Language model guided concept bottlenecks for interpretable image classification. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 19187–19197, 2023.

[44] Zhiyuan You, Lei Cui, Yujun Shen, Kai Yang, Xin Lu, Yu Zheng, and Xinyi Le. A unified model for multi-class anomaly detection. *Advances in Neural Information Processing Systems*, 35:4571–4584, 2022.

[45] 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.

[46] 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.

[47] 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.

[48] Jiaqi Zhu, Shaofeng Cai, Fang Deng, Beng Chin Ooi, and Junran Wu. Do llms understand visual anomalies? uncovering llm's capabilities in zero-shot anomaly detection. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pages 48–57, 2024.

8 Supplementary Material

9 Prompts Details

9.1 Concept Extraction Prompt

We fabricate the prompt by adding context cues that can guide the model in extracting meaningful concepts: we specify which object is present in the pictures, whether it is anomalous or not and, if so, which defect is present. Next, we employ a Chain-of-Thought prompting strategy and formulate two consecutive prompts:

1. *"Provide a description of the image that includes information about all the relevant features that are visible. Focus only on what can be seen, avoiding speculations or assumptions".*

2. *"Provided the following description, extract the five most meaningful concepts. Concepts should be defined in such a way that, observing the picture, it is possible to clearly answer with yes or no about its presence".*

It is worth mentioning that this step is applied to a small subset of the dataset, corresponding to 5% of it, so it requires the availability of only a few labeled anomalous images. When labeled anomalous samples are not available, the same procedure can be applied to synthetically generated pictures, keeping in mind that the quality of the extracted concepts is bounded by the quality of the generation process.

9.2 Dataset Annotation Prompt

A very similar prompt for dataset annotation as we did for concept extraction is employed: for each image, we include information about the object and, if present, the type of anomaly. *"I provide an image of a {category}. The image has been classified as {label}, [which implies that it shows a visible defect or anomaly, specifically {defect type}.] Knowing this, choose among the following list of concepts which ones can be clearly seen in the picture. Output the result as a JSON object of the following form: {Concept 1: True, Concept 2: False, ...}".*
Through the previous prompt, we ensure to focus only on binary concepts.

9.3 Anomaly Generation Prompt

We generate anomalous images leveraging the capabilities of Google's Nano Banana. We assume we have a set of normal pictures and a list of defects that we want to include in the synthetic images, provided by a domain expert. Next, we query a Large Language Model (LLM), in our case GPT-5, to provide ten synonyms of the defect we want to add, to ensure richer variability, and build prompts according to this structure:
"Modify this image by adding a {type of defect} to the {name of the object}, keeping the same angle, view, and pose."
Some categories and defect types, which proved to be particularly challenging for the VLM, required manually tuning the previous prompt and discarding some of the generated images. Some examples of the synthetic pictures obtained can be found in Fig. 6.

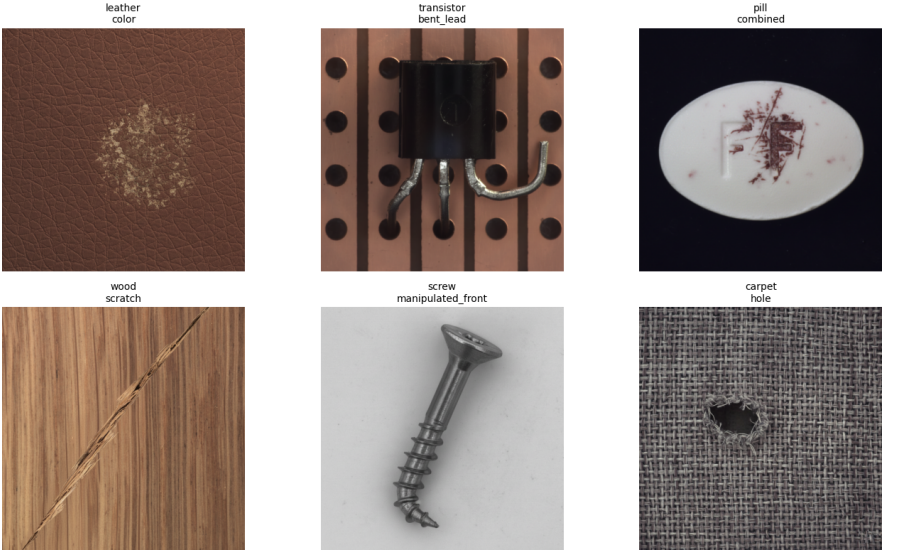


Figure 6: Examples of well-generated synthetic anomalous images.

10 Evaluation of the Concept Annotation Pipeline

In addition to the metrics reported in the main body (91.3% accuracy achieved by the automated annotation pipeline against a subset of 400 manually-annotated images), we also evaluate the full automatically annotated dataset of the *hazelnut* category against a ground-truth version of it, also computing precision and recall to make a more detailed analysis. Table 4 displays a summary of our findings. Overall, predicted concepts can be considered of good quality, especially those that appear more often in normal images; as for anomalous concepts, we observe a very high variability in terms of precision and recall, which is mainly related to those concepts that describe a diffused feature of the image (e.g., *surface discontinuity, uneven tone, etc.*), which are more difficult to capture for a VLM and can be arbitrary for a domain expert as well.

	All	Anomaly Concepts	Normal Concepts
Acc	0.93 ± 0.08	0.92 ± 0.08	0.98 ± 0.01
Prec	0.76 ± 0.24	0.71 ± 0.23	0.99 ± 0.00
Rec	0.77 ± 0.28	0.72 ± 0.28	0.99 ± 0.01

Table 4: Average value and standard deviation of accuracy, precision and recall of predicted concepts over the *hazelnut* category.

11 Additional Training Details

Each model is trained for a maximum of 100 epochs, with an early stopping mechanism triggered after ten epochs. When performance reaches a plateau after five epochs, we also

Model type	C-AUC	I-AUC
Joint	0.86	0.97
Sequential	0.85	0.95
Independent	0.87	0.96

Table 5: Comparison of the results of the three CBM learning paradigms over the MVTec AD dataset, in terms of concept prediction performance and anomaly detection.

allow the learning rate to be reduced by a factor of 10.

Transformation-based Data Augmentation. We apply transformations to enhance the model generalization ability, carefully choosing those that preserve the presence of each concept, ensuring the integrity of attribute labels. The transformations applied are the following: horizontal and vertical flips with probability 0.5, random rotation by a 25-degree angle, brightness jitter by a factor of 0.2, and contrast jitter by a factor of 0.2.

CBM Training. We attach k parallel linear layers to the CNN feature extractor that act as concept predictors, while a simple Feed-Forward Neural Network with eight neurons is employed for anomaly detection. The feature extractor undergoes a pre-training phase directly on the dataset of interest, in which it is optimized according to a multi-class classification objective to learn which object is depicted in the image, while its last layers are fine-tuned during training for the concept prediction task.

12 CBM training Modalities

Together with the joint paradigm described in the main paper, a CBM can be trained using two alternative modalities.

Independent training: the concept extractor is trained to predict concepts from inputs:

$$\hat{g} = \arg\min_{g} \sum_{i,j} \mathcal{L}_{C_j}(g_j(\mathbf{x}^i), c_j^i), \tag{12}$$

where $\mathcal{L}_{C_j}$ denotes the loss associated with the j-th concept prediction. Separately, the label predictor is trained using the ground-truth concept vectors $\mathbf{c}$ as inputs.

$$\hat{f} = \arg\min_{f} \sum_{i} \mathcal{L}_Y(f(\mathbf{c}^i), y^i) \tag{13}$$

where $\mathcal{L}_Y$ is the loss for the downstream label prediction. Note that, during deployment, $\hat{f}$ takes as input the predicted concept vector $\hat{\mathbf{c}} = \hat{g}(\mathbf{x})$ rather than the true ones. This may lead to a shift between the training and inference input distributions, potentially degrading performance.

Sequential training: first, g is trained as in the independent setting; then, f is trained directly on these predicted concepts $\hat{\mathbf{c}} = \hat{g}(\mathbf{x})$ rather than the ground truth ones.

Table 5 displays a comparison of the results obtained by the three CBM paradigms on MVTec AD. In consideration of these findings, the CBM model was trained using the Joint paradigm, since it performs similar to the Independent approach, but with a slightly better I-AUC.

	Rank		Isotonic Reg.	
Fusion	**AUROC**	**Agree**	**AUROC**	**Agree**
Max	0.81	**87.31%**	0.76	84.62%
Mean	0.88	82.95%	0.87	73.29%
Learned	0.90	79.49%	0.87	73.28%
Learned+C	**0.91**	76.52%	0.88	78.50%

Table 6: Ablation study on calibration (columns) and fusion (rows) strategies in the unsupervised setting on MVTec AD.

13 Ablation Studies on Fusion and Calibration Strategies

We conducted an ablation study on MVTec AD (unsupervised setting) analyzing (i) score calibration for the visual and concept branches and (ii) score fusion. Results are reported in Table 6, including agreement with the concept branch (*agree*), i.e., the percentage of samples whose combined-score binary prediction matches the concept branch.

(i) **Calibration:** We normalize visual and concept anomaly scores into percentile ranks of their empirical test distributions, mapping both to a common scale without supervision. Compared with supervised isotonic regression (trained on a small held-out set), rank-based normalization consistently performs better.

(ii) **Fusion:** We compare the proposed max operator against the arithmetic mean, a logistic regression on the two calibrated scores, and a logistic regression that additionally uses concept probabilities. Although the learned fusion strategies achieve higher AUROC, they consistently reduce agreement with the concept branch. Since explanations are only faithful to the concept branch, we retain the max operator and leave improved fusion strategies for future work.

14 Additional User Study Details

Participants (32 grad students/researchers, no domain expertise required) received samples via a deterministic per-participant seed with inverse-frequency sampling balanced across method, dataset, category, and label, yielding mostly disjoint stimuli across methods (7.7% image reuse) and each method's natural correct/incorrect ratio (unenforced). CONVAD's helpfulness advantage over AnomalyGPT is significant (χ^2, $p<0.001$; participant-cluster bootstrap 95% CI excludes zero), while it is on par with Anomaly-OV ($p=0.36$). We acknowledge two limitations: heatmaps are shown only for CONVAD/AnomalyGPT (Anomaly-OV produces none), and stimuli are not matched across methods.

15 Intervention Procedure

In Sec. 6.3, we demonstrate the impact on performance of manually modifying predicted concepts with their ground-truth value during inference. However, this procedure can be costly, as it requires the supervision of a human expert, so several strategies have been devised to minimize such costs by focusing first on the concepts that should be more important

to provide an increase in performance. We followed the UCP (Uncertain Concept Prediction) heuristic proposed in [32] and computed the entropy-based uncertainty related to each concept prediction:

$$\mathcal{H}(c_i) = -(p_i \cdot \log(p_i) + (1 - p_i) \cdot \log(1 - p_i)). \tag{14}$$

We then sorted the entropy scores in descending order, following the idea that concepts predicted with more uncertainty might confuse the model and lead to incorrect predictions. As mentioned in 3, since the joint model uses the predicted concept logits to perform the main task, we substitute them with either the 5th or the 95th percentile of the training distribution.

16 Concept Logits Analysis

In the unsupervised setting, where the model is trained exclusively on synthetic anomalies and evaluated on real ones, some categories achieve high concept-level AUC (C-AUC) and image-level AUC (I-AUC), while others fail to transfer effectively to real data. Two extreme examples illustrating this contrast are hazelnut, which transfers well, and metal nut, which does not. We examine the concept-logit embeddings extracted from the training and test examples to investigate this discrepancy. The test set contains only real normal and real anomalous examples, while the training set may include synthetic images in the SAG scenarios. All models compared within each category use the same random seed, training data, and evaluation set, with the latter containing only real anomalies that were not used for weak supervision in any of the models.

A well-learned bottleneck should ensure (i) a separation between normal and anomalous samples, provided that the concept set is sufficiently predictive, and (ii) tight clustering of images sharing the same or similar defect type (*e.g.* "hole" or "crack" in hazelnut). Moreover, if the synthetic anomalies are sufficiently well aligned with the real anomaly domain, their concept-logit representations should be consistent with those of real anomalies, indicating an effective transfer. To obtain qualitative insights about these aspects, we plot 2-dimensional t-SNE visualizations of the concept logits.

In the hazelnut SAG scenario, we obtain an average C-AUC of 0.89 and an I-AUC of 0.99. Fig. 7 shows a clear separation between `Train Normal` and `Train Anomalous` (synthetic) samples. `Test Normal` samples appear in the same region as `Train Normal`, which is expected since their domains match, as both correspond to real images. In addition, `Test Anomalous` (real images) are mapped to the same broad regions as synthetic anomalies (`Train Anomalous`), which is ideal. Within this anomalous region, the "print" defect forms a distinct sub-cluster, while "crack", "cut" and "hole" share more overlapping embeddings, reflecting their visual similarity. These patterns are consistent with the high AUC scores and indicate that the *synthetic anomalies generated for the hazelnut category effectively capture relevant characteristics of real anomalies*.

However, for the metal nut in the SAG scenario, the C-AUC drops to an average of 0.59, and the I-AUC drops to 0.66. In this case, in Fig. 8(a), most `Test Anomalous` (real) samples are located within regions corresponding to `Train Normal` and `Test Normal` data, rather than being mapped near the synthetic `Train Anomalous` samples. Only the "flip" defect forms a slightly separated cluster (due to its visual dissimilarity to other images), while other defects are scattered among normal samples. This qualitative mismatch between synthetic and real anomaly embeddings explains the degraded AUCs and suggests poorly generated anomalies.

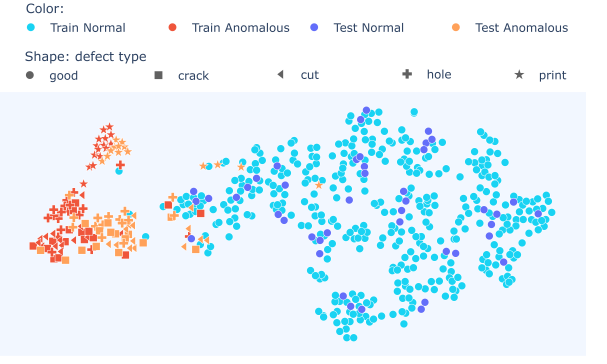
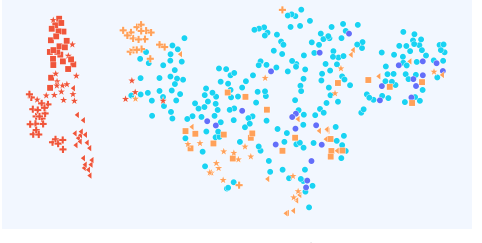
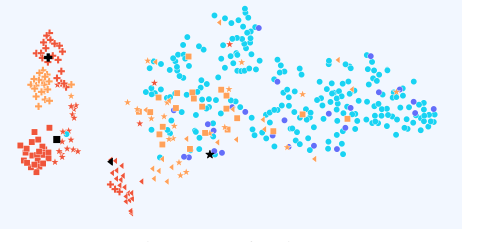


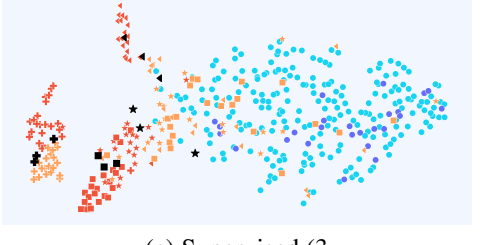
Figure 7: Hazelnut t-SNE embeddings for the SAG scenario.



(a) SAG scenario



(b) Supervised (1 anomaly-shot-per-defect)+SAG scenario



(c) Supervised (3 anomaly-shot-per-defect)+SAG scenario

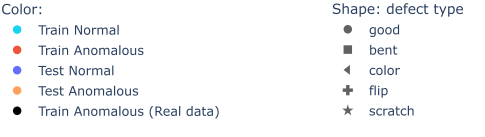


Figure 8: t-SNE representations of concept logits embeddings using `metal_nut`. Black markers highlight the few real samples used for Supervised+SAG scenarios.

Exploring the embeddings in the Supervised+SAG scenarios provides a deeper understanding of why introducing even a small number of real anomalies, combined with synthetic anomalies, reduces this discrepancy in performance. For the metal nut in the Supervised (1 anomaly-shot-per-defect)+SAG scenario (adding one real anomaly per defect type in training), the C-AUC improves to an average of 0.70 and the I-AUC to 0.84, and Fig. 8(b) shows a closer alignment between synthetic and real anomalies. With Supervised (3 anomaly-shot-per-defect)+SAG (three real anomalies per defect type, compared to one in the standard semi-supervised scenario), the C-AUC increases to 0.76 and the I-AUC to 0.89. Simultaneously, the representations of real anomalies appear closer to those of synthetic ones, and clusters associated with different defect types become more clearly separated (Fig. 8(c)). These results indicate that when using synthetic anomalies for training, even minimal real-data supervision can mitigate the domain shift and encourage the concept space to encode a more consistent defect-specific structure, eliminating the need to collect a large dataset of rare and difficult-to-acquire anomalies.

17 Additional Concept Vocabulary Analysis

We report the concept distributions for representative object categories from each of the three benchmarks (Figs. 9–12) to illustrate how the extracted concepts capture meaningful characteristics of the data. In particular, the distributions highlight clear differences between normal and anomalous samples, showing that concepts activated for specific defect types can help identify the nature of the fault from an interpretability standpoint.

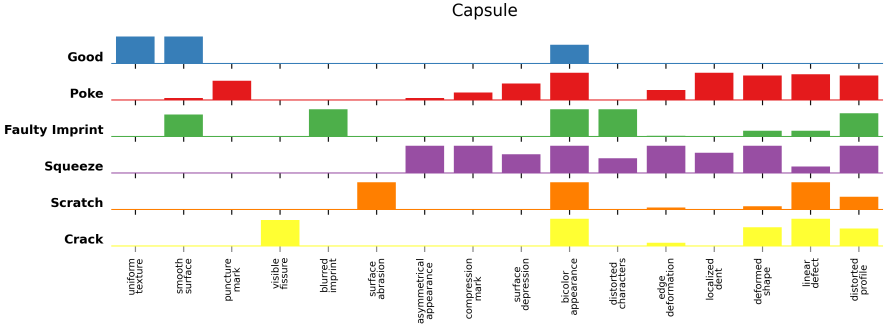


Figure 9: Concept distribution for the `capsule` category (MVTec AD).

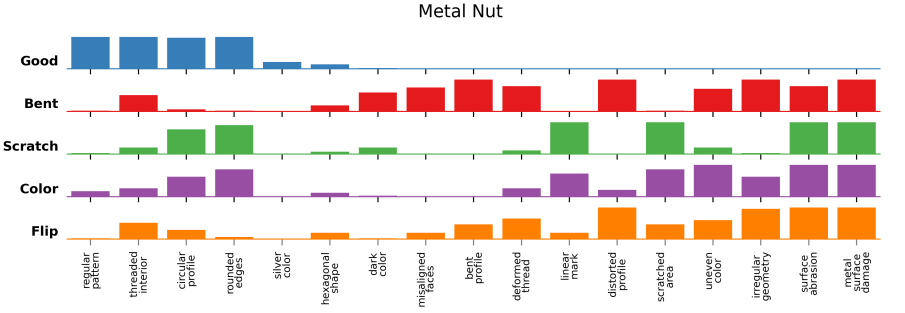


Figure 10: Concept distribution for the `metal_nut` category (MVTec AD).

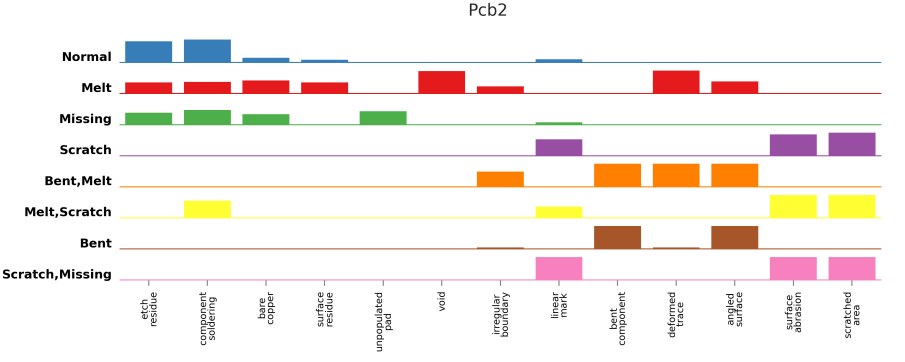


Figure 11: Concept distribution for the `pcb2` category (VisA).

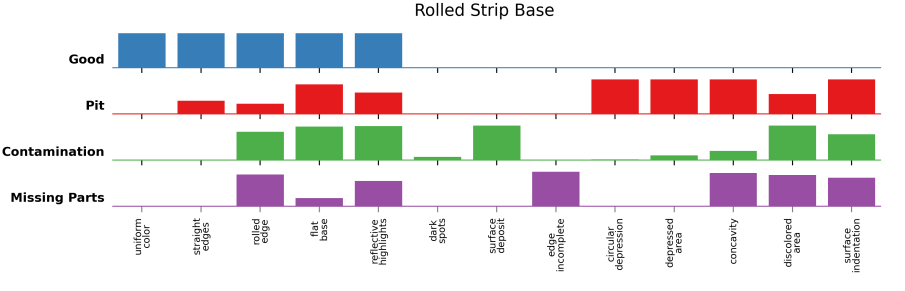


Figure 12: Concept distribution for the `rolled_strip_base` category (Real-IAD).