Long-Tailed Anomaly Detection with Learnable Class Names

Chih-Hui Ho[1] Kuan-Chuan Peng[2] Nuno Vasconcelos[1]
[1]University of California, San Diego [2]Mitsubishi Electric Research Laboratories (MERL)
chh279@ucsd.edu kpeng@merl.com nvasconcelos@ucsd.edu

Abstract

Anomaly detection (AD) aims to identify defective images and localize their defects (if any). Ideally, AD models should be able to detect defects over many image classes; without relying on hard-coded class names that can be uninformative or inconsistent across datasets; learn without anomaly supervision; and be robust to the long-tailed distributions of real-world applications. To address these challenges, we formulate the problem of long-tailed AD by introducing several datasets with different levels of class imbalance and metrics for performance evaluation. We then propose a novel method, LTAD, to detect defects from multiple and long-tailed classes, without relying on dataset class names. LTAD combines AD by reconstruction and semantic AD modules. AD by reconstruction is implemented with a transformer-based reconstruction module. Semantic AD is implemented with a binary classifier, which relies on learned pseudo class names and a pretrained foundation model. These modules are learned over two phases. Phase 1 learns the pseudo-class names and a variational autoencoder (VAE) for feature synthesis that augments the training data to combat long-tails. Phase 2 then learns the parameters of the reconstruction and classification modules of LTAD. Extensive experiments using the proposed long-tailed datasets show that LTAD substantially outperforms the state-of-the-art methods for most forms of dataset imbalance. The long-tailed dataset split is available at https://zenodo.org/records/10854201.

1. Introduction

Anomaly detection (AD) is an important problem for many manufacturing settings [23, 50, 60, 63]. To reflect practical manufacturing constraints, most datasets [5, 83, 89] are curated under the unsupervised AD setting, where no defect images are available for training. Various methods [19, 20, 43, 60, 69] have shown that this problem can be solved with high accuracy; *e.g.*, [3, 20, 31, 36, 41, 44, 69, 77, 79, 84] have success rates >95% for anomaly detection and localization on the MVTec dataset [5]. However, as illustrated in Fig. 1, these methods require a different model per image category. which compromises scalability to many classes. Recently, there has been interest in more efficient methods that use a single model to detect anomalies in all object classes [9, 14, 26, 30, 32, 76, 87]. These methods can be grouped according to the level of image semantics where they operate. On one hand, *AD by reconstruction* methods [76, 87] use a reconstruction module to project the input image into the manifold of normal images. The difference between the image and its projection is then used to detect possible defects. On the other hand, *semantic AD* methods [9, 26, 32] build explicit models of normal/abnormal images. Given the absence of abnormal training data, this is done by leveraging the knowledge of visual-language foundation models [25, 39, 57, 67]. Abnormal regions are detected using the predefined text prompts for normal and abnormal plus an image class name, *e.g.*, “a normal photo of a [CLASS]” and “an abnormal photo of a [CLASS],” where [CLASS] is a class name in the dataset, *e.g.*, “bottle” in the MVTec dataset.

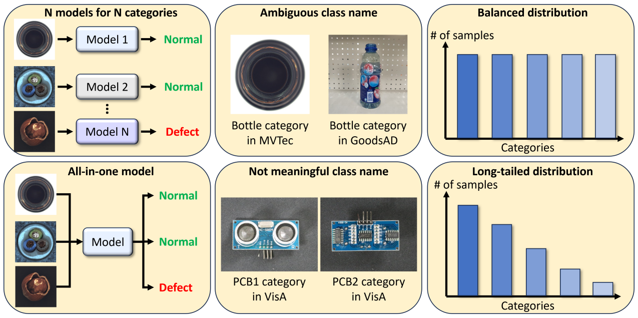


Figure 1. Challenges of long-tailed AD include (Left) designing a single model to detect anomalies over multiple image classes, (Middle) uninformative class names, and (Right) long-tailed data distributions.

The two types of methods have limitations. Reconstruction methods require modeling the complex manifold, especially for problems requiring many classes. Even when trained on large datasets, the distance between the input image and this manifold can be smaller for certain anomalies. The foundation model used by semantic AD methods can provide additional clarity, because it enables framing AD as a binary classification problem. However, this is difficult when the dataset class names are ambiguous or unknown to the foundation model. Fig. 1 shows an example of ambiguity due to the fact that the class name “bottle” refers to visually different concepts in MVTec [5] (where it means “bottle bottom”) and the GoodsAD dataset [83] (where it means “bottle side”). Hence, in MVTec, the “bottle” label may not be the most informative for the foundation model, which may associate the images with alternative labels, *e.g.* "black sphere." Sometimes, class names can be simply unknown to the foundation model, *e.g.* the classes "PCB1" and "PCB2" also shown in Fig. 1. This suggests that the foundation model should learn what are the class names that best align with these images. When this is difficult, semantic AD might benefit from the flexibility of the AD by reconstruction methods, which are not constrained by class names. Hence, this work investigates the design of AD methods that combine both AD by reconstruction and semantic AD.

Beyond this, it remains unclear whether the resulting models generalize to the *long-tailed* setting [2, 18, 29, 34, 88] where, as illustrated in Fig. 1, the sample distribution is skewed. This is particularly important because long-tailed distributions are natural in manufacturing, where different objects can have very different popularity. We formulate the problem of *long-tailed AD* by introducing long-tailed datasets, which are obtained by resampling current AD benchmarks with different imbalance factors and types of imbalance. We also propose a set of performance metrics for the long-tailed setting.

To address the challenges above, we propose a new method, LTAD, which combines AD by reconstruction and semantic AD. AD by reconstruction is implemented by combining the ALIGN [33] image encoder and a transformer-based reconstruction module (RM), trained to project image patches into the manifold of normal images. An anomaly score is then obtained by computing the difference between the input image and the result of this projection. The latent patch representation produced by the ALIGN image encoder is also mapped to the feature space of the ALIGN text encoder, to enable the implementation of semantic AD. For this, a binary normal/abnormal classifier is implemented in ALIGN text space, by using as classifier weight vectors the ALIGN representation of text prompts for "normal" and "abnormal." The posterior probability of the abnormal class, under this classifier is then used as a second anomaly score. Anomalies are detected with a combination of the two AD scores.

To address the long-tailed setting, we propose a preliminary training phase for data augmentation. This consists of learning a VAE [37, 38], which is then used to synthesize features. To make these class sensitive, the VAE is conditioned by the text encoding of the [CLASS] name, according to the ALIGN model. However, to address the ambiguity of class names, a set of learnable [CLASS] prompts are learned by backpropagation, during VAE training. In a second training phase, a mix of real and synthetic examples is used to train the LTAD model. Since there are no training anomalies, these are simulated by adding noise to the synthesized features during this stage.

Overall, we make the following contributions:

1. We show that prior methods do not perform well on long-tailed setting and formulate long-tailed AD based on 3 datasets with 9 imbalance settings and performance metrics.
2. We propose LTAD, which combines AD by reconstruction and semantic AD, performs multi-class AD, and overcomes dataset [CLASS] name ambiguity by learning names consistent with the semantic space of the ALIGN model.
3. We propose a new training strategy for LTAD which uses a novel data-augmentation procedure to address the data scarcity of long-tailed data, and learn [CLASS] names.
4. We show that LTAD outperforms the SOTA methods on the long-tailed AD. Extensive ablations confirm the efficacy of the various LTAD modules, showing that LTAD generalizes across various datasets and imbalance configurations.

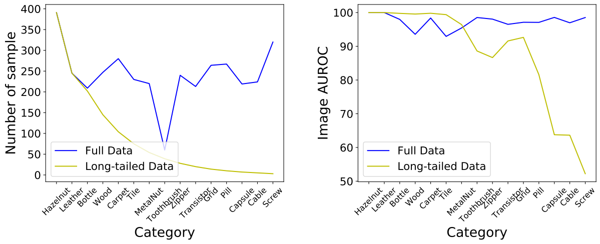
Unsupervised AD Method Conditions	**Unsupervised AD Categories**			
	$\mathcal{C}_0$	$\mathcal{C}_1$	$\mathcal{C}_2$	LTAD **(ours)**
Single model for all classes	✗	✓	✓	✓
No class name prior	✓	✓	✗	✓
Designed for the long-tailed setting	✗	✗	✗	✓
Learnable class names	✗	✗	✗	✓

Table 1. LTAD addresses some important challenges for real-world AD, previously not considered in (*e.g.*, $\mathcal{C}_0$: [16, 19, 20, 31, 43, 54, 59–61, 64, 69, 77, 79–81, 84], $\mathcal{C}_1$: [26, 30, 49, 76, 87], $\mathcal{C}_2$: [9, 10, 14, 32, 82]).

2. Related works

Unsupervised anomaly detection (AD): Unsupervised AD aims to identify defective images and localize the defects without observing any defect images during training. Tab. 1 groups recent AD methods into 3 categories. **Category** $\mathcal{C}_0$ contains earlier works [16, 19, 20, 31, 43, 60, 69, 84] that use a different model per image category. Student-teacher methods [20, 69, 84] use a pretrained teacher encoder and a student encoder optimized to match its predictions. AD is based on the difference of their predictions. Flow based methods [59, 61, 77] fit a Gaussian distribution to the feature vectors of normal images and use out-of-distribution criteria to perform AD. Reconstruction based methods [54, 79–81] train models to reconstruct normal samples and use reconstruction error for AD. Some other methods in category $\mathcal{C}_0$ are discussed in the AD survey papers [11, 12, 47, 65]. More parameter efficient than these methods, LTAD only needs a single model for all the classes instead of one model per class.

This is the setting adopted by the more recent methods of **Category** $\mathcal{C}_1$ [26, 30, 49, 76, 87]. For example, UniAD [76] improves prior reconstruction-based methods by using a neighborhood attention mask to avoid information leak, while AnomalyGPT [26] uses a large vision-language model to provide explanations for defective regions. While typically leveraging foundation models, these methods do not use the class name to detect anomalies. This akin to LTAD but unlike methods in **category** $\mathcal{C}_2$ [9, 10, 14, 32, 82], which also use a single model for all classes but require class names. For example, WinCLIP [32] uses the CLIP [57] model to compute the anomaly score, by measuring similarity between image and text feature vectors for several predefined normal/abnormal text prompts. Some works in categories $\mathcal{C}_1$ and $\mathcal{C}_2$ [14, 26] also leverage auxiliary training data (*e.g.*, train on VisA [89] and test on MVTec [5]), a setting that is not considered in this work.



(a) dataset distribution (b) anomaly detection performance

Figure 2. Preliminary study with UniAD on MVTec. Image classes (x-axis) are sorted by popularity. (a) Dataset distribution of MVTec vs. long-tailed version. (b) AD performance on the two datasets.

More importantly, all the prior AD works assume balanced datasets [5, 70, 83, 89], where the number of samples is relatively balanced across classes. However, this is an unlikely setting for real-world applications where different objects tend to have different popularity. As summarized in Table 1, this work addresses the combinatorial challenge of imbalanced training set, absent class name, and single model for multiple object classes.

Long-tailed recognition: Real-world data distributions are often imbalanced across classes. Prior works in classification [18, 34, 40, 45, 72] have shown that the data imbalance degrades performance for minority, or tail, classes. Long-tailed recognition methods aim to avoid this. Prior long-tailed recognition methods related to this work can be mainly categorized into (1) data re-sampling, (2) loss re-weighting, and (3) representation learning. Data re-sampling methods [6, 13, 15, 22, 27, 28, 75] balance the sample distribution by under-sampling majority classes or augmenting minority classes. For example, [2, 29, 40, 45, 88] utilize a generative model to synthesize samples or features from minority classes. Loss re-weighting methods [7, 18, 34, 46, 58, 66, 71] weigh the loss function by class cardinality, usually assigning higher weights for loss terms dependent on minority classes. Representation learning methods [21, 35, 74] focus on learning a more powerful feature encoder. For example, [21, 51, 52] leverage the generalizable knowledge from large foundation models [57]. For a more detailed review, please see the recent survey papers [73, 85, 86]. While prior long-tailed works focus on image classification, we are the first to investigate the long-tailed setting for AD.

3. Long-tailed anomaly detection

Motivation: Previous AD works assume that different image classes are equally populated. However, in most industrial applications, different objects have different costs, production schedules, *etc*. This creates long-tailed distributions where certain classes have much higher example cardinality than others. Extensive research in areas like classification, indicates that systems not trained to account for this class imbalance tend to overfit on popular classes and ignore the less popular ones [18, 34, 34, 40, 45, 58, 71]. To test whether this also holds for AD, we perform some preliminary experiments, using the MVTec dataset and a long-tailed version, obtained by image resampling. The sample distributions of the two datasets are shown in Fig. 2(a). Fig. 2(b) compares the performance on the two datasets of UniAD [76], one of the best open-source AD methods based on a single model that detects multi-class anomalies. The figure confirms that UniAD performs well on MVTec but degrades considerably for the long-tailed version, where it overfits to majority classes and severely underperforms on minority ones. This experiment highlights the need for the methods that explicitly address the long-tailed AD problem. This requires datasets and performance metrics, which we discuss next.

Dataset	Max Class Sample	Imbalanced Factor β
MVTec [5]	391	{100, 200}
VisA [89]	905	{100, 200, 500}
DAGM [70]	1000	{50, 100, 200, 500}

Table 2. The statistics of the long-tailed splits we use across the 3 datasets. For all datasets, we consider both the *exp* and *step* imbalance.

Dataset collection: Following [8], given a balanced dataset, we create a long-tailed version by sampling the training set, while the test set remains unchanged. The distribution of the sampled training set depends on two factors: the imbalance factor β and the imbalance type. β is the ratio between the cardinalities of the most and least populated classes, *i.e.*, $\beta = \frac{\max_c\{N_c\}}{\min_c\{N_c\}}$ where N_c is the number of samples of class c. To prevent overfitting on a specific imbalance distribution, we consider two types of imbalance: exponential (*exp*) and step (*step*). While the former indicates that N_c decays exponentially across classes c, the latter indicates a binary split into majority classes of size $\max\{N_c\}$ and minority classes of size $\min\{N_c\}$. Individual long-tailed datasets are denoted by type and imbalance factor. For example, *exp*100 indicates a training set exponentially imbalanced with $\beta = 100$. We define the half most/least populated classes as majority/minority classes and construct the long-tailed dataset by randomly sampling from the original dataset, without repetition. When the number of samples of a class is less than the desired number of samples, all samples are kept.

Tasks and metrics: Both anomaly detection (AD) and anomaly segmentation (AS) are considered. Following [26, 30, 76, 79], we use the Area Under the Receiver Operating Curve (AUROC) at image and pixel level for AD and AS, respectively. We report two types of results. The first is average performance (Avg), which is the performance averaged across all classes. The second is the pair of average performance for majority classes (High) and average performance for minority classes (Low).

Datasets: We consider 3 datasets: MVTec [5], VisA [89] and DAGM [70]. Various long-tailed datasets are built from each. Tab. 2 shows the maximum number of samples across classes and the imbalance factors considered per dataset. For all datasets, we consider both the *exp* and *step* imbalance types. Our proposed dataset splits will be released upon publication.

4. The `LTAD` anomaly score

In this section, we propose a novel method, `LTAD`, for the unsupervised long-tailed anomaly detection task. As shown in

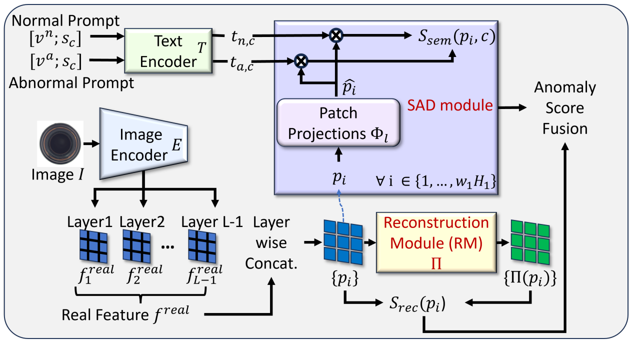


Figure 3. The LTAD architecture combines AD by reconstruction and semantic AD scores ($\mathcal{S}_{rec}$ and $\mathcal{S}_{sem}$, respectively), implemented by the RM and SAD modules. We use an image E and text encoder T from a pretrained foundation model to extract images features f^{real} and text features $t_{n,c}$,$t_{a,c}$ derived from text-prompts that include a static component to discriminate between normal (v^n) and abnormal (v^n) and a learned component s_c to make this discrimination class-sensitive.

Fig. 3, LTAD uses a combination of (1) AD by reconstruction and (2) semantic AD. AD by reconstruction is a common AD approach, where a model is trained to reconstruct normal images [1, 4, 17, 48, 62, 76, 78]. At inference, this model projects abnormal images into the normal image manifold. AD can thus be implemented by thresholding the magnitude of the reconstruction error. Semantic AD explicitly trains a classifier to discriminate between normal and abnormal images. This is less commonly used, since abnormal images are not available in the training set. LTAD overcomes this challenge by leveraging the understanding of the "abnormal" concept by the pretrained ALIGN [33] foundation model and a learned semantic descriptor for the classes in the training set. The implementation of these modules is as follows.

AD by reconstruction. The reconstruction module (RM) $\Pi(.)$ is a transformer [42, 53, 56, 76] trained to reconstruct the features extracted from the image I by a pretrained encoder E of L layers. Given an image $I \in \mathbb{R}^{W\times H\times 3}$ from class $c \in \mathcal{C}$, E extracts feature tensor $f_l^{real} \in \mathbb{R}^{W_l\times H_l\times C_l}$ from layer $l \in \{1...L\}$. Since the feature tensor from the last layer f_L^{real} represents the global semantics of I, it tends to degrade the AD performance [30, 60, 76], which requires local semantics. Hence, to perform AD, this tensor is dropped and the first $L-1$ tensors $\{f_l^{real}\}_{l=1}^{L-1}$ are remapped to the dimensions of f_1^{real} (*i.e.*, $W_1 \times H_1$) by bilinear interpolation along the spatial dimension. In the following, we use the notation f_l^{real} to represent this interpolated version and define $f^{real} = [f_1^{real}; \dots; f_{L-1}^{real}]$ as the feature tensor extracted across the $L-1$ layers. This tensor is then split into $W_1 \times H_1$ patch feature vectors $\{p_i\}_{i=1}^{W_1\times H_1}$, which are fed as tokens to the RM transformer $\Pi(.)$. Given patch i, the anomaly score of the AD by reconstruction module is the squared error

$$\mathcal{S}_{rec}(p_i) = ||\Pi(p_i) - p_i||^2. \tag{1}$$

Semantic AD (SAD). The goal of semantic AD is two-fold: 1) to give the anomaly detector sensitivity to normal/abnormal classes, and 2) to leverage the prior knowledge about normality/abnormality available in a large foundation model. This allows the AD to discriminate between the two conditions without requiring abnormal images for training. As shown in Fig. 3, the semantic AD module is a binary classifier of a projection $\widehat{p}_i$ of patch p_i into normal/abnormal classes. The layer-wise components p_{il} of the patch feature vector p_i are first projected into vectors $\Phi_l(p_{il})$ with the dimension d of the text embedding of the ALIGN model. These projections are implemented by projection modules $\Phi_l : \mathbb{R}^{C_l} \rightarrow \mathbb{R}^d$, $l = 1, \dots, L-1$, where each Φ_l is implemented with a linear layer. The layer-wise features are then aggregated into a single patch feature vector

$$\widehat{p}_i = \max_l(\{\Phi_l(p_{il})\}_l) \tag{2}$$

by max pooling over layers. The resulting vector $\widehat{p}_i$ is then fed to a binary classifier of parameters $t_{n,c}$ (normal) and $t_{a,c}$ (abnormal), where c is the class of image I, which computes the posterior probability of an anomaly using a softmax layer with temperature scaling τ

$$\mathcal{S}_{sem}(p_i, c) = \frac{\exp(t_{a,c} \bullet \widehat{p}_i/\tau)}{\exp(t_{n,c} \bullet \widehat{p}_i/\tau) + \exp(t_{a,c} \bullet \widehat{p}_i/\tau)}, \tag{3}$$

where "$\bullet$" denotes the dot product. This is used as the semantic AD score for the images of class c.

The main difficulty of this process is to learn the classifier parameters $t_{n,c}$, $t_{a,c}$ without explicit supervision, since there are no training images of anomalies. To overcome this problem we leverage the prior for normal/abnormal classification provided by the ALIGN model. This is implemented by feeding to ALIGN a normal text prompt v_n and an abnormal text prompt v_a that apply to all classes. While we have considered several possibilities (see Tab. 9), the best AD performance was achieved by setting v_n ="a" and v_a ="a broken." Unless otherwise noted, we use these prompts in what follows. To further make the anomaly score sensitive to the image semantics, this is complemented by an image class prompt s_c. Unlike prior works [9, 14, 32] that assume the access to the class names (*e.g.*, "bottle," or "hazelnut" in MVTec [5]), we assume that the class name is unknown. This is important to support the classes that are unknown to the ALIGN model or even to most humans, such as "PCB1" vs. "PCB2" in Fig. 1. Instead, inspired by [24], we use a *pseudo* class name s_c learned per class c. This is implemented by prompting the text encoder T with a prompt s_c per class c, and learning prompts s_c as discussed below. The resulting set of *semantic sensitive AD prompts* $\mathcal{P} = \{[v^n;\, s_c],\, [v^a;\, s_c]\}_c$ is mapped to a set of classifier parameters $\{(t_{n,c},\, t_{a,c})\}_c$ by the text encoder T of the ALIGN model, according to

$$t_{n,c} = T([v^n;\, s_c]) \qquad t_{a,c} = T([v^a;\, s_c]). \tag{4}$$

LTAD score: The overall anomaly score of a patch feature p_i of class c is defined as the linear combination

$$\mathcal{S}(p_i, c) = \mathcal{S}_{rec}(p_i) + \lambda \mathcal{S}_{sem}(p_i,\, c), \tag{5}$$

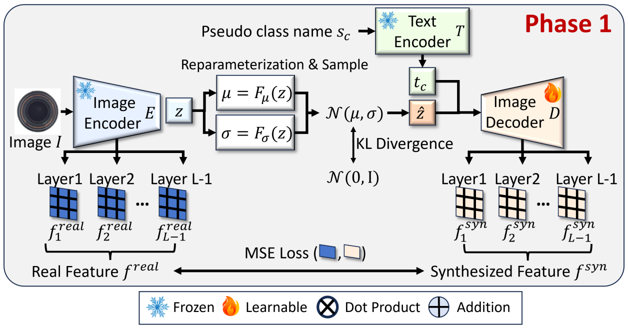


Figure 4. Phase 1 of LTAD training learns a VAE-style decoder D for feature augmentation conditioned on a learned pseudo class name s_c.

where λ is the hyperparameter to balance $\mathcal{S}_{rec}(p_i)$ and $\mathcal{S}_{sem}(p_i, c)$ such that both scores have comparable ranges.

5. Training

The training of LTAD is divided into two phases.

5.1. Phase 1: Class sensitive data augmentation

This training phase seeks two goals: to 1) overcome the data scarcity of long-tailed AD, by augmenting the training set with normal examples of minority classes and abnormal examples of all classes, 2) learn the class sensitive prompts s_c required by the semantic AD score of (3)-(4).

Fig. 4 summarizes this training procedure. Given an image $I \in \mathbb{R}^{W\times H\times 3}$ of class $c \in \mathcal{C}$, the pretrained encoder E extracts the feature tensor $f^{real} = [f_1^{real}; \ldots; f_{L-1}^{real}]$ plus a latent code $z = f_L^{real}$, which is the feature vector from the last encoder layer (L). An image decoder D, whose architecture is the mirror copy of E, is then trained to sample corresponding feature vectors, using a procedure inspired by the variational autoencoder (VAE) [37, 38]. A latent feature $\widehat{z}$ is sampled from a normal distribution $\mathcal{N}(\mu, \sigma)$ of parameters $\mu = F_\mu(z)$ and $\sigma = F_\sigma(z)$, where F_μ and F_σ are learned linear transformations. The decoder D then synthesizes a feature tensor from $\widehat{z}$.

In the long-tailed setting, the performance of D degrades for classes c with few training images. To ameliorate this problem, D is conditioned by the prior knowledge about the class, in the form of a text-derived prototype feature vector t_c that represents class c for feature synthesis. This is obtained by prompting the text encoder of ALIGN with the pseudo-class name s_c, *i.e.*, $t_c = T(s_c)$. The feature prototype is then concatenated with the image-dependent latent feature $\widehat{z}$ to create the input to D, which finally synthesizes a feature tensor $\{f_l^{syn}\}_{l=1}^{L-1} = D(\widehat{z}, t_c)$ of dimensions equal to those of f_l^{real}.

Following standard practices for VAE training, D and s_c are learned by optimizing a loss function

$$\mathcal{L}_{\mathbb{P}_1} = \frac{1}{L-1}\sum_{l=1}^{L-1} ||f_l^{syn} - f_l^{real}||^2 - KL(\mathcal{N}(\widehat{z}-\mu, \sigma)||\mathcal{N}(0, I)), \tag{6}$$

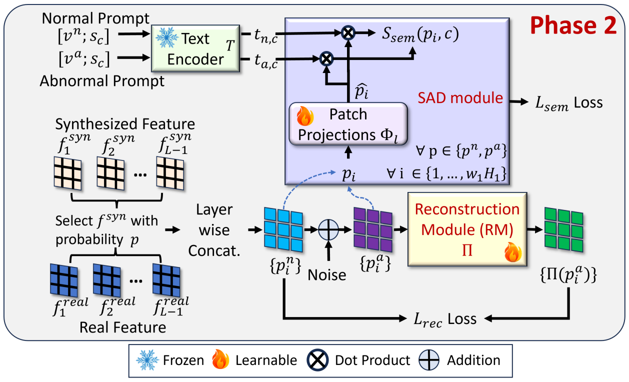


Figure 5. Phase 2 of LTAD training learns the parameters of the reconstruction module (RM) and patch projections Φ_l that map visual features into the semantic space of the semantic AD (SAD) module.

that combines the reconstruction mean square error (MSE), a regularization constraint based on the Kullback-Leibler divergence (KL) that encourages a normal distribution, and the reparametrization trick of [37]. The text T and image E encoders are those of the pretrained ALIGN model and kept frozen throughout training. Note that this process encourages the simultaneous satisfaction of multiple goals: 1) learning a decoder that can be used to synthesize features from the tail classes, 2) align these features with the semantic representation t_c produced by the text encoder of ALIGN, and 3) improve the quality of feature synthesis for tail classes, by leveraging this alignment. After training, the learned prompts s_c are used in (4).

5.2. Data augmentation

When this training phase is completed, the decoder D works as a data augmentation device, producing synthetic feature tensors f^{syn} in the semantic neighborhood of a feature tensor f^{real} extracted from a real image. This is used to augment the training data in an online fashion during the second phase of training. Two types of data augmentation are considered.

Long-tailed classes. To counteract the imbalanced nature of long-tailed datasets, data augmentation is implemented by selecting the real f^{real} or synthetic f^{syn} feature vectors with probabilities $(p_c,\ 1-p_c)$ respectively. The selected feature vector f is split into $W_1 \times H_1$ patch feature vectors $\{p_i^n\}_{i=1}^{W_1\times H_1}$ where the n superscript denotes that these are normal features.

Anomalies. To counteract the lack of anomalies during training, random noise (sampled from normal distribution) is added to normal patch features p_i^n to produce pseudo-anomaly patch features p_i^a, as in [76]. This process is repeated for all normal patches during training. No random noise is added during inference.

5.3. Phase 2: Anomaly detection

Data augmentation is used in the second phase of training to learn the parameters of 1) the RM transformer $\Pi(.)$ used to reconstruct features and 2) the modules $\Phi_l(.)$ used in (2) to project patch features into the semantic space of ALIGN.

Reconstruction Module (RM): As shown in the bottom right of Fig. 5, the RM $\Pi(.)$ is trained to project the pseudo anomaly patch features p_i^a into the reconstructed patch features $\Pi(p_i^a)$ in the manifold of the normal patch features p_i^n. This is implemented with the RM transformer to minimize the loss

$$\mathcal{L}_{rec} = \frac{1}{W_1 H_1} \sum_{i=1}^{W_1 H_1} ||\Pi(p_i^a) - p_i^n||^2. \tag{7}$$

Semantic patch projections: As shown in the top right of Fig. 5, the functions Φ_l compute projections of the patches p_i into the semantic space of ALIGN, where the classifier parameters of (4) are defined. The functions Φ_l are trained to encourage the alignment between the projected patch features and the text features by minimizing the binary cross entropy loss

$$\mathcal{L}_{sem}(c) = \frac{-1}{W_1 H_1} \sum_{i=1}^{W_1 H_1} y_i \log(\mathcal{S}_{sem}(p_i, c)), y_i = \begin{cases} 1, \text{ if } p_i = p_i^a \\ 0, \text{ if } p_i = p_i^n, \end{cases} \tag{8}$$

where c is the image class and $\mathcal{S}_{sem}(.)$ is the semantic score of (3). Note that both phase 1 and phase 2 share the same text encoder T, which is fixed in both phases. The total loss function for phase 2 is $\mathcal{L}_{\mathbb{P}_2} = \mathcal{L}_{rec} + \mathcal{L}_{sem}(c)$.

6. Experiments

In this section, we report on various experiments designed to evaluate LTAD. While we compute average performance across majority (High), minority (Low), and all (All) classes in all experiments, we omit the High and Low values in some cases, for brevity. A complete listing is provided in the supplement.

Baselines: For a given training configuration (*e.g.*, *exp*100), the same training set is used for all the baselines [26, 43, 64, 76, 79] and LTAD. For baselines other than RegAD [30], their official code and training/testing setting are used. RegAD is trained on all classes, instead of its leave-one-class-out setting, for fair comparison. During testing, we use 2 training examples per class as the support set to estimate the normal distribution. Since RegAD requires a support set of 2 images per class, it is not applicable to some configurations (*e.g.*, *exp*200 and *step*200 in MVTec) where only 1 image is available for some minority classes. Cut & Paste [43] supports anomaly detection but not localization.

Training details: To train LTAD, we use the pretrained visual language foundation model ALIGN [33]. Each input image is scaled to 224×224 and the features f^{real} are extracted from layers $\{3, 10, 17, 37\}$ of the ALIGN image encoder. By default, the length of the pseudo class name is set to 2 and initialized with the text "object object." In phase 2, the probability p_c of selecting real features is 0.5. The hyperparameter λ in (5) is set to 500, 400, and 300 for MVtec, VisA, and DAGM, respectively. Unless otherwise noted, we use $v^n =$ "a" and $v^a =$ "a broken" in (4) and $\tau = 1$ in (3). Phase 1 is trained for 100 epochs and phase 2 for 500 epochs, both using the AdamW optimizer with the learning rate 1e-4. Pytorch [55] is used for implementation.

6.1. Comparisons to the state-of-the-art

Tab. 3-5 summarize the AD and AS performance of all the methods on MVTec, VisA, and DAGM, respectively. The best and second best performances for each dataset configuration are highlighted in **bold** and underline, respectively. On MVTec, LTAD is compared to 6 baselines under 4 different configurations. As shown in Tab. 3, early methods, such as Cut & Paste [43], MKD [64], and DRAEM [79], are not suitable to detect and localize defects across classes with a single model, leading to inferior performance. While more recent models [26, 30, 76] can detect defects across classes, they do not perform well across levels of dataset imbalance. LTAD is less affected by skewed distributions and outperforms most baselines in all tasks. The only exceptions are RegAD, which outperforms LTAD in 2 of the 8 tasks, and UniAD in 1 of the 8 tasks. Tab. 4-5 show that the gains of LTAD over the baselines are even larger for VisA and DAGM, where it achieves the best performance in all tasks. Overall, LTAD outperforms the baselines on 29 out of the 32 (90.6%) tasks defined by the 3 datasets.

Fig. 7 shows a comparative visualization of anomaly detections by UniAD and LTAD. Note how the detections of LTAD are much more localized and selective of the anomaly. Additional visualizations are provided in the supplement.

Beyond LTAD, the performance of LTAD without the semantic AD module is shown in the penultimate columns of Tab. 3-5. Even without this module, LTAD beats the baselines on 28 out of 32 (87.5%) tasks. However, by adding the semantic AD module, performance improves on 31 out of 32 (96.9%) tasks, highlighting the efficacy of the semantic AD module.

6.2. Ablation study

We ablate different designs of LTAD in Tab. 6-8, where the experiments are denoted by the experiment ID (expID).

Ablation study of AD by reconstruction: Fig. 6 and Tab. 6 compare different RM designs (*i.e.*, LTAD without the semantic AD module) on the MVTec *step*100 dataset. Fig. 6 studies the effect of the length of the pseudo class name prompt on AD performance, showing that the performance peaks for length 2. Tab. 6 summarizes the detection and segmentation performance of several variants of LTAD for majority (High), minority (Low) and all (All) classes. expID 6.2 first shows that replacing the EfficientNet [68] features of UniAD by those of ALIGN increases both AD and AS performance by 2.38% and 1.44%, respectively. In expID 6.3, we replace the text conditioned VAE of LTAD by an autoencoder trained on ALIGN features, to generate features for phase 1 of LTAD training. This underperforms LTAD (expID 6.8) by 2.57% (0.6%) for AD (AS), highlighting the efficacy of the VAE of LTAD. In expID 6.4, we replace the probability $p_c = 0.5$, used for synthetic feature selection in phase 2 of LTAD, by a more sophisticated design, where p_c is inversely proportional to the class cardinality (*i.e.*, the classes with less samples have higher probability of selecting synthesized features). When compared to LTAD (expID 6.8),

Config.	Task	Cut & Paste	MKD	DRAEM	RegAD	UniAD	AnomalyGPT	LTAD w/o SAD	LTAD
*exp*100	Det.	75.89	78.92	79.57	82.43	87.70	87.44	88.74	**88.86**
	Seg.	N/A	85.95	85.17	**95.20**	93.95	89.68	94.00	94.46
*exp*200	Det.	75.07	79.93	78.82	N/A	86.21	85.80	**86.94**	86.05
	Seg.	N/A	86.01	82.95	N/A	93.26	90.15	93.40	**94.18**
*step*100	Det.	76.57	79.61	69.82	81.54	83.37	85.95	87.05	**87.36**
	Seg.	N/A	85.90	79.65	**95.10**	91.47	89.28	93.13	93.83
*step*200	Det.	76.53	79.31	71.64	N/A	81.32	82.47	85.33	**85.60**
	Seg.	N/A	86.03	76.79	N/A	89.29	89.45	91.78	**92.12**

Table 3. Quantitative comparison on MVTec dataset.

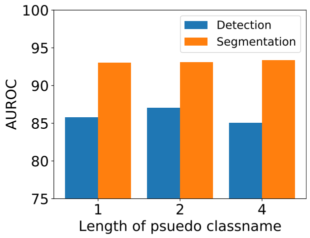


Figure 6. Ablation on the length of pseudo class name.

Config.	Task	RegAD	UniAD	AnomalyGPT	LTAD w/o SAD	LTAD
*exp*100	Det.	71.36	77.31	70.34	79.27	**80.00**
	Seg.	94.40	95.03	80.32	95.07	**95.56**
*exp*200	Det.	72.10	76.87	69.78	78.55	**80.21**
	Seg.	94.69	94.80	79.48	94.51	**95.36**
*exp*500	Det.	N/A	73.67	68.18	77.25	**78.53**
	Seg.	N/A	94.35	78.83	94.04	**94.66**
*step*100	Det.	71.80	78.83	71.98	82.80	**84.80**
	Seg.	94.99	96.04	82.30	96.16	**96.57**
*step*200	Det.	71.65	77.64	69.78	83.79	**84.03**
	Seg.	94.52	95.66	81.97	95.89	**96.27**
*step*500	Det.	N/A	71.84	62.88	82.42	**83.33**
	Seg.	N/A	95.03	81.48	95.50	**96.41**

Table 4. Quantitative comparison on VisA dataset.

Config.	Task	RegAD	UniAD	AnomalyGPT	LTAD w/o SAD	LTAD
*exp*100	Det.	84.86	84.34	85.31	93.35	**94.40**
	Seg.	90.29	90.13	77.20	96.93	**97.30**
*exp*200	Det.	84.86	83.56	83.29	92.83	**94.29**
	Seg.	90.29	89.73	77.16	96.16	**97.19**
*exp*500	Det.	84.86	81.35	83.47	92.08	**93.54**
	Seg.	90.29	88.63	76.87	95.99	**97.01**
*step*100	Det.	84.86	81.11	86.48	91.94	**93.97**
	Seg.	90.28	89.11	78.76	96.38	**97.07**
*step*200	Det.	84.86	80.33	84.73	91.78	**93.79**
	Seg.	90.29	89.07	78.29	96.04	**96.84**
*step*500	Det.	84.86	80.04	85.08	91.82	**92.78**
	Seg.	90.29	88.53	78.75	95.64	**96.65**

Table 5. Quantitative comparison on DAGM dataset.

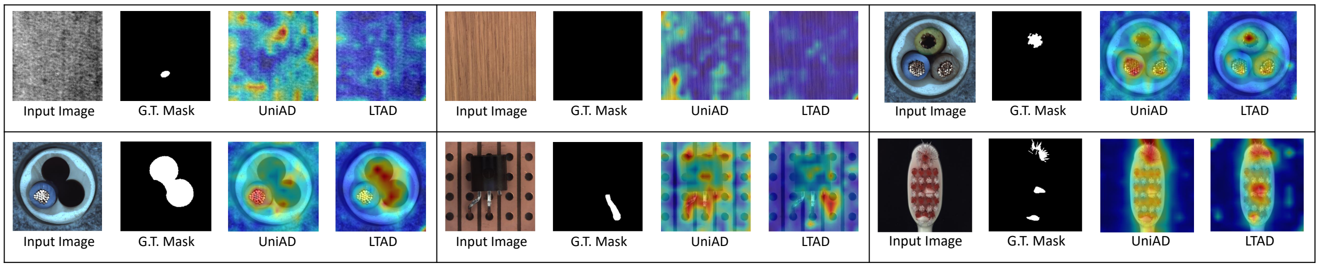


Figure 7. Qualitative comparison of UniAD and LTAD anomalies. The images without defects (*i.e*., normal image) have a black ground truth mask.

there is no benefit in adopting this distribution aware sampling (DAS) mechanism. The remaining experiments then ablate different possibilities for the class prompt s_c. Experiments expID 6.5 and expID 6.6 consider the case where the class names are available in the dataset. In expID 6.5, the name of class c is simply used as s_c. This underperforms all other experiments, where s_c is a learned pseudo class name. These experiments vary on the procedure used to initialize s_c for learning. ExpID 6.6, which uses the class name in the dataset as initialization, outperforms expID 6.7, which randomly initializes s_c. Altogether, these results support the claim that, while the class specific prompts are important, not all the class names are informative and the prompts should be learned. Finally, all methods underperform LTAD (expID 6.8), which initializes all s_c with the word "object." This initialization also outperformed various other words that we have tried. The configuration of expID 6.8 is used as the default for all other experiments in the paper.

Ablation study of Semantic AD: Tab. 7 ablates different SAD module designs (*i.e*., LTAD without the RM module) on the MVTec *step*100 dataset. ExpID 7.1 replaces (2) by a direct projection of the concatenated vectors p_i into the d-dimensional text space, using a linear transformation $\widehat{\Phi} : \mathbb{R}^{\sum_{l=1}^{L-1} C_l} \rightarrow \mathbb{R}^d$. When compared to LTAD (ExpID 7.4), this degrades AS performance significantly, most likely due to a greater difficulty in accounting for the different resolutions of features from different layers. The remaining experiments compare the implementation of (2) with different pooling operations, namely max vs. mean. Experiments expID 7.2 and 7.4 show that mean pooling is better than no layer aware projection, but inferior to the max pooling of LTAD. Finally, expID 7.3 uses max pooling and investigates the use of multiple word-prompts v^n and v^a in (4). This is inspired by the gains reported for ensembling

expID	method	use ALIGN	AE or VAE	use DAS	learned s_c	s_c init.	Detection			Segmentation		
							All	High	Low	All	High	Low
6.1	UniAD	✗	N/A	N/A	N/A	N/A	82.63	**99.60**	67.79	91.47	**96.15**	87.38
6.2	UniAD	✓	N/A	N/A	N/A	N/A	85.01	99.39	72.43	92.91	95.99	90.20
6.3	LTAD	✓	AE	✗	✗	N/A	84.48	99.13	71.67	92.53	94.77	90.57
6.4	LTAD	✓	VAE	✓	✓	"object"	86.06	97.77	75.81	92.99	94.44	**91.73**
6.5	LTAD	✓	VAE	✗	✗	class name	85.39	99.11	73.39	92.57	94.56	90.82
6.6	LTAD	✓	VAE	✗	✓	class name	86.12	99.07	74.80	92.94	94.60	91.49
6.7	LTAD	✓	VAE	✗	✓	random	85.95	99.01	74.53	92.92	95.07	91.03
6.8	LTAD	✓	VAE	✗	✓	"object"	**87.05**	99.07	**76.54**	**93.13**	95.07	91.44

Table 6. Ablation Study without the SAD module on MVTec-*step*100. Acronyms: DAS: distribution aware sampling; init.: initialization.

expID	Layer Aware Projection	Pooling Operation	Prompt	Detection			Segmentation		
				All	High	Low	All	High	Low
7.1	✗	N/A	Single	83.78	96.35	72.79	87.18	91.08	83.77
7.2	✓	Mean	Single	81.25	90.95	72.76	88.28	91.89	85.12
7.3	✓	Max	Multiple	77.19	94.06	62.43	91.14	**95.37**	87.44
7.4 (`LTAD`)	✓	Max	Single	**84.12**	**97.02**	**72.84**	**91.36**	95.13	**88.07**

Table 7. Ablation Study without RM on MVTec-*step*100 dataset.

expID	assign $s_{c=i}$ to class i	use text encoder T	Detection			Segmentation		
			All	High	Low	All	High	Low
8.1	✗	✓	72.76	81.06	65.49	63.74	62.09	65.18
8.2	✓	✗	59.79	63.34	56.69	69.83	70.95	68.85
8.3 (`LTAD`)	✓	✓	**84.12**	**97.02**	**72.84**	**91.36**	**95.13**	**88.07**

Table 8. Importance of pseudo class name s_c on MVTec-*step*100.

	v^a			
v^n	a broken	a damaged	an abnormal	a defective
a	**84.12** / 91.36	82.95 / 91.70	82.20 / 91.33	83.66 / **91.87**
a normal	83.71 / 91.39	82.74 / 91.21	83.47 / 91.23	82.94 / 91.26
a good	75.68 / 90.75	82.14 / 91.22	81.03 / 91.15	82.09 / 91.23
a flawless	65.63 / 87.61	79.09 / 91.00	76.13 / 90.24	83.89 / 91.42

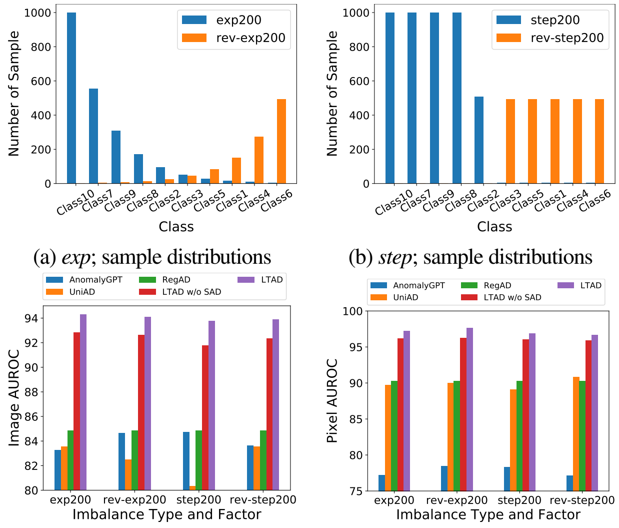
Table 9. Ablation on different normal/abnormal text prompts (*i.e.*, v^n and v^a) on MVTec *step*100. The AD/AS performances are reported.

multiple prompts when visual language foundation models are used for open-set classification [57]. Comparing experiments expID 7.3 and 7.4 shows that there is no similar advantage for semantic AD. Note that jointly using 2 anomaly scores on MVTec *step*100 (See Tab. 3 *step*100) outperforms using either the reconstruction by AD of expID 6.8 or the semantic AD of expID 7.4 solely, indicating both modules contribute to the gain.

Tab. 8 ablates the importance of the learned pseudo class name s_c for the semantic AD module, showing that it does not solely rely on the normal/abnormal prompt for anomaly detection. ExpID 8.1 shuffles the pseudo class names across classes by assigning $s_{c=i}$, the pseudo class name of class i, to class j, where $i \neq j$. Compared to `LTAD` (expID 8.3), this hurts performance significantly. ExpID 8.2 tests the alternative of eliminating the text encoder of ALIGN completely, simply learning a binary classifier of weight vectors $t_{n,c}$, $t_{a,c}$ per image class c. This approach is even less effective, showing the importance of the prior knowledge encoded in the foundation model about both classes and normal/abnormal images. The small sizes of the AD datasets are not sufficient to overcome the use of this prior.

Tab. 9 further ablates different combinations for normal v^n and abnormal v^a text prompts. The combinations of ("a", "a broken") and ("a", "a defective") outperform all others. Note that a poor choice of these prompts can degrade performance, although the latter seems to be more sensitive to the choice of normal than abnormal prompt. This shows the importance of leveraging the prior knowledge of foundation model about the two conditions.

Distribution imbalance: A set of experiments were performed on DAGM [70] to evaluate the robustness of `LTAD` to the class imbalance of the training dataset. To avoid the possibility of overfitting on a specific class order, we repeated these experiment with reverse class order (the least populated classes become the most populated ones). We considered both exponential and step dataset imbalance. Fig. 8 (a,b) show the class cardinalities before (blue) and after (orange) reversing the class order. Fig. 8 (c-d) compare the AD and AS performance of the different approaches. For all dataset configurations, `LTAD` outperforms the baselines by at least 9.1% (5.9%) for AD (AS). Part of these gains are due to SAD module, which improves performance by 1.62% (1.01%) for AD (AS) on average. These results show that `LTAD` generalizes across imbalance factors, that SAD module consistently improves performance, and that its gains are insensitive to class order.



(a) *exp*; sample distributions

(b) *step*; sample distributions

(c) detection performance

(d) segmentation performance

Figure 8. Top: Sample distributions before and after class reversal, for exponential and step imbalance, respectively. Bottom: AS and AD AUROC vs. imbalance for different dataset configurations.

7. Conclusion

In this work, we have introduced the task of long-tailed AD, by proposing datasets and performance metrics and a novel AD method, `LTAD`, tailored for the long-tailed setting. `LTAD` detect defects from multiple and long-tailed classes, without relying on dataset class names. It combines AD by reconstruction and semantic AD modules. AD by reconstruction is implemented with a transformer-based reconstruction module. Semantic AD is implemented with a binary classifier, which relies on learned pseudo class names and a pretrained foundation model. These modules are learned over two phases. Phase 1 learns the pseudo-class names and a VAE for feature synthesis that augments the training data to combat long-tails. Phase 2 then learns the parameters of the reconstruction and classification modules of `LTAD`. Experiments show that `LTAD` outperforms the SOTA AD methods on most long-tailed datasets considered and all the components of `LTAD` contribute to its superior performance.

Acknowledgement CH and KCP were supported by Mitsubishi Electric Research Laboratories. CH and NV were partially funded by NSF awards IIS-2303153 and gift from Qualcomm.

References

[1] Samet Akçay, Amir Atapour-Abarghouei, and T. Breckon. GANomaly: Semi-supervised anomaly detection via adversarial training. *ArXiv*, abs/1805.06725, 2018. 4

[2] Gundeep Arora, Vinay Kumar Verma, Ashish Mishra, and Piyush Rai. Generalized zero-shot learning via synthesized examples. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4281–4289, 2017. 2, 3

[3] Jaehyeok Bae, Jaehyeon Lee, and Seyun Kim. PNI: Industrial anomaly detection using position and neighborhood information. 2022. 1

[4] Paul Bergmann, Sindy Löwe, Michael Fauser, David Sattlegger, and Carsten Steger. Improving unsupervised defect segmentation by applying structural similarity to autoencoders. *ArXiv*, abs/1807.02011, 2018. 4

[5] 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. 1, 2, 3, 4

[6] Mateusz Buda, Atsuto Maki, and Maciej A. Mazurowski. A systematic study of the class imbalance problem in convolutional neural networks. *Neural networks: the official journal of the International Neural Network Society*, 106:249–259, 2017. 3

[7] Kaidi Cao, Colin Wei, Adrien Gaidon, Nikos Aréchiga, and Tengyu Ma. Learning imbalanced datasets with label-distribution-aware margin loss. In *Neural Information Processing Systems*, 2019. 3

[8] Kaidi Cao, Colin Wei, Adrien Gaidon, Nikos Arechiga, and Tengyu Ma. Learning imbalanced datasets with label-distribution-aware margin loss. In *Advances in Neural Information Processing Systems*, 2019. 3

[9] Yunkang Cao, Xiaohao Xu, Chen Sun, Yuqi Cheng, Zongwei Du, Liang Gao, and Weiming Shen. Segment any anomaly without training via hybrid prompt regularization. (arXiv:2305.10724). 1, 2, 4

[10] Yunkang Cao, Xiaohao Xu, Chen Sun, Xiaonan Huang, and Weiming Shen. Towards generic anomaly detection and understanding: Large-scale visual-linguistic model (GPT-4V) takes the lead. *ArXiv*, abs/2311.02782, 2023. 2

[11] Raghavendra Chalapathy and Sanjay Chawla. Deep learning for anomaly detection: A survey. *ArXiv*, abs/1901.03407, 2019. 2

[12] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM Comput. Surv.*, 41(3), 2009. 2

[13] N. Chawla, K. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. SMOTE: Synthetic minority over-sampling technique. *ArXiv*, abs/1106.1813, 2002. 3

[14] Xuhai Chen, Yue Han, and Jiangning Zhang. A zero-/few-shot anomaly classification and segmentation method for CVPR 2023 VAND workshop challenge tracks 1&2: 1st place on zero-shot AD and 4th place on few-shot AD. *ArXiv*, abs/2305.17382, 2023. 1, 2, 4

[15] Peng Chu, Xiao Bian, Shaopeng Liu, and Haibin Ling. Feature space augmentation for long-tailed data. In *European Conference on Computer Vision*, 2020. 3

[16] Niv Cohen and Yedid Hoshen. Sub-image anomaly detection with deep pyramid correspondences. *ArXiv*, abs/2005.02357, 2020. 2

[17] Anne Collin and Christophe De Vleeschouwer. Improved anomaly detection by training an autoencoder with skip connections on images corrupted with stain-shaped noise. *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 7915–7922, 2020. 4

[18] Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge J. Belongie. Class-balanced loss based on effective number of samples. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9260–9269, 2019. 2, 3

[19] Thomas Defard, Aleksandr Setkov, Angélique Loesch, and Romaric Audigier. PaDiM: A patch distribution modeling framework for anomaly detection and localization. In *ICPR Workshops*, 2020. 1, 2

[20] Hanqiu Deng and Xingyu Li. Anomaly detection via reverse distillation from one-class embedding. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9727–9736, 2022. 1, 2

[21] Bowen Dong, Pan Zhou, Shuicheng Yan, and Wangmeng Zuo. LPT: Long-tailed prompt tuning for image classification. In *The Eleventh International Conference on Learning Representations*, 2023. 3

[22] Chris Drummond. C4.5, class imbalance, and cost sensitivity: Why under-sampling beats oversampling. 2003. 3

[23] Yehonatan Fridman, Matan Rusanovsky, and Gal Oren. ChangeChip: A reference-based unsupervised change detection for PCB defect detection. *2021 IEEE Physical Assurance and Inspection of Electronics (PAINE)*, pages 1–8, 2021. 1

[24] Rinon Gal, Yuval Alaluf, Yuval Atzmon, Or Patashnik, Amit Haim Bermano, Gal Chechik, and Daniel Cohen-or. An image is worth one word: Personalizing text-to-image generation using textual inversion. In *The Eleventh International Conference on Learning Representations*, 2023. 4

[25] Rohit Girdhar, Alaaeldin El-Nouby, Zhuang Liu, Mannat Singh, Kalyan Vasudev Alwala, Armand Joulin, and Ishan Misra. Imagebind one embedding space to bind them all. *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15180–15190, 2023. 1

[26] Zhaopeng Gu, Bingke Zhu, Guibo Zhu, Yingying Chen, Ming Tang, and Jinqiao Wang. AnomalyGPT: Detecting industrial anomalies using large vision-language models. *arXiv preprint arXiv:2308.15366*, 2023. 1, 2, 3, 6

[27] Hui Han, Wenyuan Wang, and Binghuan Mao. Borderline-SMOTE: A new over-sampling method in imbalanced data sets learning. In *International Conference on Intelligent Computing*, 2005. 3

[28] Haibo He and Edwardo A. Garcia. Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering*, 21:1263–1284, 2009. 3

[29] Haibo He, Yang Bai, Edwardo A. Garcia, and Shutao Li. ADASYN: Adaptive synthetic sampling approach for imbalanced learning. *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*, pages 1322–1328, 2008. 2, 3

[30] Chaoqin Huang, Haoyan Guan, Aofan Jiang, Ya Zhang, Michael Spratlin, and Yanfeng Wang. Registration based few-shot anomaly detection. In *European Conference on Computer Vision (ECCV)*, 2022. 1, 2, 3, 4, 6

[31] Jeeho Hyun, Sangyun Kim, Giyoung Jeon, Seungwook Kim, Kyunghoon Bae, and Byungjin Kang. ReConPatch: Contrastive patch representation learning for industrial anomaly detection. *ArXiv*, abs/2305.16713, 2023. 1, 2

[32] Jongheon Jeong, Yang Zou, Taewan Kim, Dongqing Zhang, Avinash Ravichandran, and Onkar Dabeer. WinCLIP: Zero-/few-shot anomaly classification and segmentation. *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 19606–19616, 2023. 1, 2, 4

[33] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc V. Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *International Conference on Machine Learning*, 2021. 2, 4, 6

[34] Bingyi Kang, Saining Xie, Marcus Rohrbach, Zhicheng Yan, Albert Gordo, Jiashi Feng, and Yannis Kalantidis. Decoupling representation and classifier for long-tailed recognition. *ArXiv*, abs/1910.09217, 2019. 2, 3

[35] Bingyi Kang, Yu Li, Sa Xie, Zehuan Yuan, and Jiashi Feng. Exploring balanced feature spaces for representation learning. In *International Conference on Learning Representations*, 2021. 3

[36] Donghyeon Kim, Chaewon Park, Suhwan Cho, and Sangyoun Lee. FAPM: Fast adaptive patch memory for real-time industrial anomaly detection. *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5, 2022. 1

[37] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. *CoRR*, abs/1312.6114, 2013. 2, 5

[38] Diederik P. Kingma and Max Welling. An introduction to variational autoencoders. *ArXiv*, abs/1906.02691, 2019. 2, 5

[39] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross B. Girshick. Segment anything. *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023. 1

[40] Adit Krishnan, Ashish Sharma, Aravind Sankar, and H. Sundaram. An adversarial approach to improve long-tail performance in neural collaborative filtering. *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, 2018. 3

[41] Sungwook Lee, Seunghyun Lee, and Byung Cheol Song. CFA: Coupled-hypersphere-based feature adaptation for target-oriented anomaly localization. *IEEE Access*, 10:78446–78454, 2022. 1

[42] Yunseung Lee and Pilsung Kang. AnoViT: Unsupervised anomaly detection and localization with vision transformer-based encoder-decoder. *IEEE Access*, PP:1–1, 2022. 4

[43] Chun-Liang Li, Kihyuk Sohn, Jinsung Yoon, and Tomas Pfister. CutPaste: Self-supervised learning for anomaly detection and localization. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9659–9669, 2021. 1, 2, 6

[44] Hanxi Li, Jianfei Hu, Bo Li, Hao Chen, Yongbin Zheng, and Chunhua Shen. Target before shooting: Accurate anomaly detection and localization under one millisecond via cascade patch retrieval. *ArXiv*, abs/2308.06748, 2023. 1

[45] Shuang Li, Kaixiong Gong, Chi Harold Liu, Yulin Wang, Feng Qiao, and Xinjing Cheng. MetaSAug: Meta semantic augmentation for long-tailed visual recognition. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5208–5217, 2021. 3

[46] 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. 3

[47] Jiaqi Liu, Guoyang Xie, Jingbao Wang, Shangwen Li, Chengjie Wang, Feng Zheng, and Yaochu Jin. Deep industrial image anomaly detection: A survey. *ArXiv*, abs/2301.11514, 2023. 2

[48] Wenqian Liu, Runze Li, Meng Zheng, Srikrishna Karanam, Ziyan Wu, Bir Bhanu, Richard J. Radke, and Octavia I. Camps. Towards visually explaining variational autoencoders. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8639–8648, 2019. 4

[49] 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. *ArXiv*, abs/2310.14228, 2023. 2

[50] Jianhong Ma, Tao Zhang, Cong Yang, Yangjie Cao, Lipeng Xie, Hui Tian, and Xuexiang Li. Review of wafer surface defect detection methods. *Electronics*, 12(8), 2023. 1

[51] Teli Ma, Shijie Geng, Mengmeng Wang, Jing Shao, Jiasen Lu, Hongsheng Li, Peng Gao, and Y. Qiao. A simple long-tailed recognition baseline via vision-language model. *ArXiv*, abs/2111.14745, 2021. 3

[52] Teli Ma, Shijie Geng, Mengmeng Wang, Sheng Xu, Hongsheng Li, Baochang Zhang, Peng Gao, and Yu Qiao. Unleashing the potential of vision-language models for long-tailed visual recognition. In *33rd British Machine Vision Conference 2022, BMVC 2022, London, UK, November 21-24, 2022*. BMVA Press, 2022. 3

[53] Pankaj Mishra, Riccardo Verk, Daniele Fornasier, Claudio Piciarelli, and Gian Luca Foresti. VT-ADL: A vision transformer network for image anomaly detection and localization. *2021 IEEE 30th International Symposium on Industrial Electronics (ISIE)*, pages 01–06, 2021. 4

[54] Shancong Mou, Xiaoyi Gu, Meng Cao, Haoping Bai, Ping Huang, Jiulong Shan, and Jianjun Shi. RGI: Robust GAN-inversion for mask-free image inpainting and unsupervised pixel-wise anomaly detection. In *ICLR*, 2023. 2

[55] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An imperative style, high-performance deep learning library. In *Neural Information Processing Systems*, 2019. 6

[56] Jonathan Pirnay and Keng Yip Chai. Inpainting transformer for anomaly detection. In *International Conference on Image Analysis and Processing*, 2021. 4

[57] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, 2021. 1, 2, 3, 8

[58] Jiawei Ren, Cunjun Yu, Shunan Sheng, Xiao Ma, Haiyu Zhao, Shuai Yi, and Hongsheng Li. Balanced meta-softmax for long-tailed visual recognition. *ArXiv*, abs/2007.10740, 2020. 3

[59] Danilo Jimenez Rezende and Shakir Mohamed. Variational inference with normalizing flows. *ArXiv*, abs/1505.05770, 2015. 2

[60] Karsten Roth, Latha Pemula, Joaquin Zepeda, Bernhard Scholkopf, Thomas Brox, and Peter Gehler. Towards total recall in industrial anomaly detection. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14298–14308, 2021. 1, 2, 4

[61] Marco Rudolph, Bastian Wandt, and Bodo Rosenhahn. Same same but differnet: Semi-supervised defect detection with normalizing flows. In *Winter Conference on Applications of Computer Vision (WACV)*, 2021. 2

[62] Mohammad Sabokrou, Mohammad Khalooei, Mahmood Fathy, and Ehsan Adeli. Adversarially learned one-class classifier for novelty detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3379–3388, 2018. 4

[63] Radhwan A. A. Saleh, Mehmet Zeki Konyar, Kaplan Kaplan, and H. Metin Ertunç. Tire defect detection model using machine learning. In *2022 2nd International Conference on Emerging Smart Technologies and Applications (eSmarTA)*, pages 1–5, 2022. 1

[64] Mohammadreza Salehi, Niousha Sadjadi, Soroosh Baselizadeh, Mohammad Hossein Rohban, and Hamid R. Rabiee. Multiresolution knowledge distillation for anomaly detection. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14897–14907, 2020. 2, 6

[65] Mohammadreza Salehi, Hossein Mirzaei, Dan Hendrycks, Yixuan Li, Mohammad Hossein Rohban, and Mohammad Sabokrou. A unified survey on anomaly, novelty, open-set, and out of-distribution detection: Solutions and future challenges. *Transactions on Machine Learning Research*, 2022. 2

[66] Jun Shu, Qi Xie, Lixuan Yi, Qian Zhao, Sanping Zhou, Zongben Xu, and Deyu Meng. Push the student to learn right: Progressive gradient correcting by meta-learner on corrupted labels. *ArXiv*, abs/1902.07379, 2019. 3

[67] Yixuan Su, Tian Lan, Huayang Li, Jialu Xu, Yan Wang, and Deng Cai. PandaGPT: One model to instruction-follow them all. *ArXiv*, abs/2305.16355, 2023. 1

[68] Mingxing Tan and Quoc Le. EfficientNet: Rethinking model scaling for convolutional neural networks. In *Proceedings of the 36th International Conference on Machine Learning*, pages 6105–6114. PMLR, 2019. 6

[69] 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. 1, 2

[70] Matthias Wieler, Tobias Hahn, and Fred. A. Hamprecht. Weakly supervised learning for industrial optical inspection, 2007. https://hci.iwr.uni-heidelberg.de/node/3616. 3, 8

[71] Tong Wu, Qingqiu Huang, Ziwei Liu, Yu Wang, and Dahua Lin. Distribution-balanced loss for multi-label classification in long-tailed datasets. *ArXiv*, abs/2007.09654, 2020. 3

[72] Tz-Ying Wu, Pedro Morgado, Pei Wang, Chih-Hui Ho, and Nuno Vasconcelos. Solving long-tailed recognition with deep realistic taxonomic classifier. *ArXiv*, abs/2007.09898, 2020. 3

[73] Lu Yang, He Jiang, Qing Song, and Jun Guo. A survey on long-tailed visual recognition. *International Journal of Computer Vision*, 130:1837 – 1872, 2022. 3

[74] Yuzhe Yang and Zhi Xu. Rethinking the value of labels for improving class-imbalanced learning. *ArXiv*, abs/2006.07529, 2020. 3

[75] Bee Wah Yap, Khatijahhusna Abd Rani, Hezlin Aryani Abd Rahman, Simon James Fong, Zuraida Khairudin, and Nik Nik Abdullah. An application of oversampling, undersampling, bagging and boosting in handling imbalanced datasets. In *International Conference on Advanced Data and Information Engineering*, 2013. 3

[76] 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. Curran Associates, Inc., 2022. 1, 2, 3, 4, 5, 6

[77] Jiawei Yu1, Ye Zheng, Xiang Wang, Wei Li, Yushuang Wu, Rui Zhao, and Liwei Wu. FastFlow: Unsupervised anomaly detection and localization via 2D normalizing flows. *ArXiv*, abs/2111.07677, 2021. 1, 2

[78] M. Zaheer, Jin ha Lee, M. Astrid, and Seung-Ik Lee. Old is gold: Redefining the adversarially learned one-class classifier training paradigm. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14171–14181, 2020. 4

[79] Vitjan Zavrtanik, Matej Kristan, and Danijel Skocaj. DRAEM - A discriminatively trained reconstruction embedding for surface anomaly detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 8330–8339, 2021. 1, 2, 3, 6

[80] Gongjie Zhang, Kaiwen Cui, Tzu-Yi Hung, and Shijian Lu. Defect-GAN: High-fidelity defect synthesis for automated defect inspection. *2021 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 2523–2533, 2021.

[81] Hui Min Zhang, Z. Wang, Zuxuan Wu, and Yuwei Jiang. DiffusionAD: Denoising diffusion for anomaly detection. *ArXiv*, abs/2303.08730, 2023. 2

[82] Jiangning Zhang, Xuhai Chen, Zhucun Xue, Yabiao Wang, Chengjie Wang, and Yong Liu. Exploring grounding potential of VQA-oriented GPT-4V for zero-shot anomaly detection. *ArXiv*, abs/2311.02612, 2023. 2

[83] Jian Zhang, Runwei Ding, Miaoju Ban, and Ge Yang. PKU-GoodsAD: A supermarket goods dataset for unsupervised anomaly detection and segmentation. *ArXiv*, abs/2307.04956, 2023. 1, 3

[84] 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 (CVPR)*, pages 3914–3923, 2023. 1, 2

[85] Yifan Zhang, Bingyi Kang, Bryan Hooi, Shuicheng Yan, and Jiashi Feng. Deep long-tailed learning: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45:10795–10816, 2021. 3

[86] Yongshun Zhang, Xiu-Shen Wei, Boyan Zhou, and Jianxin Wu. Bag of tricks for long-tailed visual recognition with deep convolutional neural networks. In *AAAI*, pages 3447–3455, 2021. 3

[87] Ying Zhao. OmniAL: A unified cnn framework for unsupervised anomaly localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3924–3933, 2023. 1, 2

[88] Yang Zou, Zhiding Yu, B. V. K. Vijaya Kumar, and Jinsong Wang. Domain adaptation for semantic segmentation via class-balanced self-training. *ArXiv*, abs/1810.07911, 2018. 2, 3

[89] Yang Zou, Jongheon Jeong, Latha Pemula, Dongqing Zhang, and Onkar Dabeer. Spot-the-difference self-supervised pre-training for anomaly detection and segmentation. *arXiv preprint arXiv:2207.14315*, 2022. 1, 2, 3