MITSUBISHI ELECTRIC RESEARCH LABORATORIES
https://www.merl.com

Toward Long-Tailed Online Anomaly Detection through Class-Agnostic Concepts

Yang, Chiao-An; Peng, Kuan-Chuan; Yeh, Raymond

TR2025-124 August 26, 2025

Abstract

Anomaly detection (AD) identifies the defect regions of a given image. Recent works have studied AD, focusing on learning AD without abnormal images, with long-tailed distributed training data, and using a unified model for all classes. In addition, online AD learning has also been explored. In this work, we expand in both directions to a realistic setting by considering the novel task of long-tailed online AD (LTOAD). We first identified that the offline state- of-the-art LTAD methods cannot be directly applied to the online setting. Specifically, LTAD is class-aware, requiring class labels that are not available in the online setting. To address this challenge, we propose a class-agnostic frame- work for LTAD and then adapt it to our online learning set- ting. Our method outperforms the SOTA baselines in most offline LTAD settings, including both the industrial manufacturing and the medical domain. In particular, we ob- serve +4.63% image-AUROC on MVTec even compared to methods that have access to class labels and the number of classes. In the most challenging long-tailed online setting, we achieve +0.53% image-AUROC compared to baselines.

IEEE International Conference on Computer Vision (ICCV) 2025

Mitsubishi Electric Research Laboratories, Inc.
201 Broadway, Cambridge, Massachusetts 02139

Toward Long-Tailed Online Anomaly Detection through Class-Agnostic Concepts

Chiao-An Yang[1] Kuan-Chuan Peng[2] Raymond A. Yeh[1]
[1]Department of Computer Science, Purdue University [2]Mitsubishi Electric Research Laboratories
[1]{yang2300,rayyeh}@purdue.edu [2]kpeng@merl.com

Abstract

Anomaly detection (AD) identifies the defect regions of a given image. Recent works have studied AD, focusing on learning AD without abnormal images, with long-tailed distributed training data, and using a unified model for all classes. In addition, online AD learning has also been explored. In this work, we expand in both directions to a realistic setting by considering the novel task of long-tailed online AD (LTOAD). We first identified that the offline state-of-the-art LTAD methods cannot be directly applied to the online setting. Specifically, LTAD is class-aware, requiring class labels that are not available in the online setting. To address this challenge, we propose a class-agnostic framework for LTAD and then adapt it to our online learning setting. Our method outperforms the SOTA baselines in most offline LTAD settings, including both the industrial manufacturing and the medical domain. In particular, we observe +4.63% image-AUROC on MVTec even compared to methods that have access to class labels and the number of classes. In the most challenging long-tailed online setting, we achieve +0.53% image-AUROC compared to baselines. Our LTOAD benchmark is released here[1].

1. Introduction

Anomaly detection (AD) [8, 65, 96] is the task of identifying defect regions in a given image. This task is important due to its high valued applications in industrial manufacturing [1, 5, 18, 47, 58, 74, 86, 103] and medical [28, 42, 43, 47] settings.

Recently, LTAD [34] explored the long-tailed (LT) [92] setting of unsupervised one-class AD, where abnormal images are not given during training and the training set is unevenly distributed. This long-tailed distribution assumes that images of certain classes significantly outnumber others, which is a realistic assumption for real-world applications. Importantly, LTAD shows that prior work on evenly distributed data performs suboptimally on LT data. Meanwhile, recent works [16, 22, 55, 61, 79, 84] approach AD with online learning to leverage abnormal images during test time. However, they do not study the case where the classes follow a long-tail distribution, where the performance gap of head and tail classes in the online setting may be significant during offline training. This motivates us to study the long-tailed online AD (LTOAD) task by proposing a benchmark and a novel model.

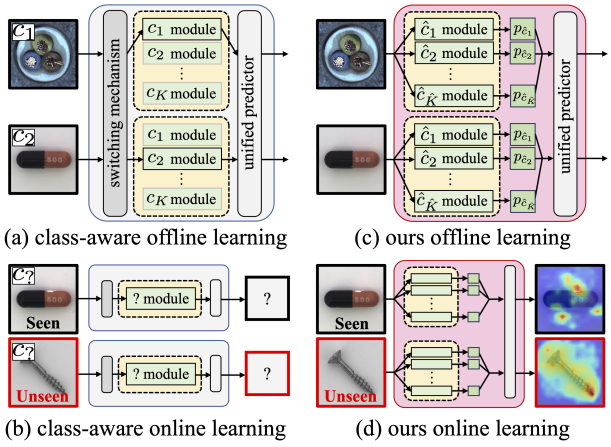
At first glance, it may seem that one can take a state-of-the-art (SOTA) long-tailed offline model, *e.g.*, LTAD [34], and combine it with online learning to solve LTOAD. However, several limitations within the LTAD's framework prevent it from being effective in the online setting. First, LTAD is a class-aware method, where they assume that the class information is given. However, class labels are not available in the online setting. Instead, a **class-agnostic** method is required as illustrated in Fig. 1. Second, LTAD's encoder-decoder architecture does not leverage the latest vector quantization VAE [20, 57, 59], which is more effective, as shown in recent work. Third, the prompt learning designs of LTAD and other AD works [7, 9, 10, 37, 40, 51, 52, 67, 100–102] ignore some aspects of abnormal prompts, *e.g.*, LTAD does not support class-specific abnormal prompts.

To address these challenges, we propose a novel model that is class-agnostic, with a vector quantized VAE, and a comprehensive prompt learning framework, for the task of long-tailed online anomaly detection. Experiments show that our method outperforms the SOTA baselines significantly by at least 4% on several detection benchmarks without access to the class labels and the number of classes.

In the following sections, we first discuss how to make existing class-aware methods class-agnostic, specifically on LTAD (Sec. 2). Next, we define the task of long-tailed online AD and present our solution (Sec. 3). Finally, we provide details of each proposed module (Sec. 4), followed by the experimental results (Sec. 5) and related works (Sec. 6).

Our contributions are summarized as follows:

[1]https://doi.org/10.5281/zenodo.16283852



(a) class-aware offline learning (c) ours offline learning

(b) class-aware online learning (d) ours online learning

Figure 1. **Comparison of LTOAD to class-aware anomaly detection methods on offline and online learning.** (a) Class-aware methods have a class-specific module for each class c in the class set $\mathcal{C}$. (b) These methods cannot work in online learning when the class labels are unavailable. (c) LTOAD solves this problem by learning a concept set $\widehat{\mathcal{C}}$ to approximate $\mathcal{C}$. We note that $\widehat{K} = |\widehat{\mathcal{C}}|$ does not need to match $K = |\mathcal{C}|$. (d) In online learning, LTOAD can weight input images of **seen** and **unseen** classes with existing concept-specific modules, *i.e.*, $\{p_{\widehat{c}}\}_{\widehat{c}\in\widehat{\mathcal{C}}}$.

- We propose the task and benchmark for **long-tailed online anomaly detection (LTOAD)**.
- We propose a class-agnostic framework for both offline and online learning that removes the constraint of requiring additional class information in previous work. Along with vector-quantization VAE and a comprehensive prompt learning scheme that benefits both offline and online learning.
- LTOAD outperforms SOTA offline long-tailed methods and online baselines on MVTec, VisA, DAGM, and Uni-Medical. It also shows promising generalizability to cross-dataset settings.

2. Making LTAD class-agnostic

2.1. Background

Anomaly Detection (AD). Given an image $\boldsymbol{X} \in \mathbb{R}^{H\times W\times 3}$ of height H and width W, an AD model $\mathtt{F}_\theta$, parametrized by θ, aims to predict an abnormal map $\widehat{\boldsymbol{Y}} \in \{0,1\}^{H\times W}$ or an abnormal label $y \in \{0,1\}$) indicating whether the image is abnormal or not, where $\widehat{\boldsymbol{Y}}_i = \mathtt{F}_\theta(\boldsymbol{X}_i)$.

Beyond the standard setting, LTAD [34] proposes a more challenging setup of unsupervised unified AD that focuses on an unevenly distributed training set. Each class c in the class set $\mathcal{C}$ belongs to either the set of "head classes" $\mathcal{C}^{\mathtt{head}}$ or "tail classes" $\mathcal{C}^{\mathtt{tail}}$ depending on the number of times a class appears in the data set. That is, images from the classes in $\mathcal{C}^{\mathtt{head}}$ significantly outnumber those from $\mathcal{C}^{\mathtt{tail}}$.

Vision and language models (VLM) in AD. Recent works [7, 9, 10, 25, 34, 37, 40, 49, 51, 52, 67, 94, 100] have shown the benefit of foundation VLM for AD. A foundation model consists of a image encoder $\mathtt{E}^{\mathtt{I}}$ and an text encoder $\mathtt{E}^{\mathtt{T}}$. The encoders encode images and text to a shared space such that cosine similarly between the embeddings captures the relationship between image and text. We use the notation $\langle\cdot\rangle$ for cosine similarity that implicitly supports broadcasting, *i.e.*, when $\langle\cdot\rangle$ is between a matrix with a vector, we broadcast on the spatial dimension.

In more detail, recent works [30, 34, 57, 62] extract fine-grained image features from intermediate layers or blocks of $\mathtt{E}^{\mathtt{I}}$. We denote these intermediate visual feature maps as $\boldsymbol{F}^{\mathtt{i}}$ and the final visual feature vectors as $\boldsymbol{f}^{\mathtt{f}}$, *i.e.*, $(\boldsymbol{F}^{\mathtt{i}}, \boldsymbol{f}^{\mathtt{f}}) = \mathtt{E}^{\mathtt{I}}(\boldsymbol{X})$. The precise descriptions of $\boldsymbol{F}^{\mathtt{i}}$ and $\boldsymbol{f}^{\mathtt{f}}$ are provided in Appendix A1.5.

Class-aware AD. Existing works [7, 9, 10, 22, 25, 34, 37, 40, 49, 51, 52, 57, 61, 67, 84, 94] consist of class-aware modules in their architecture. These works require an additional class label $c \in \mathcal{C}$ for each image as input. Specifically, they assume that the class c of the input image is also provided to the model, *i.e.*, $\widehat{\boldsymbol{Y}}_i = \mathtt{F}_\theta(\boldsymbol{X}_i, c_i)$. As depicted in Fig. 1, a hard switching mechanism is used in their framework to select the output of a specific class-aware module, *i.e.* $\boldsymbol{M}_{c_i}$, to use for each instance. In this work, we do not assume that the class label is provided, as it would typically be unavailable in the online setting. Instead, we develop a class-agnostic approach, which we describe next.

2.2. Concept set for removing class-awareness

To remove the requirement of having the class label c, we introduce a concept set $\widehat{\mathcal{C}}$ where we assume that the class information c can be represented as a composition of multiple concepts in $\widehat{\mathcal{C}}$. For example, the class 'transistor' is related to and derived from concepts 'semiconductors' and 'circuits'. In other words, for each image of class c, instead of applying a hard one-hot label, we employ a soft weighting mechanism and assign a soft label $\boldsymbol{p} \in \mathbb{R}^{\widehat{K}}$ where $\widehat{K} = |\widehat{\mathcal{C}}|$. There are a few advantages. First, we can reduce $\widehat{K}$ for model efficiency, since $\widehat{\mathcal{C}}$ does not need to match $\mathcal{C}$ exactly and can be much smaller. Next, this soft weighting mechanism is applicable when an unseen class $\notin \mathcal{C}$ appears in a realistic testing scenario, while previous class-aware works [34, 57] would fail to function.

For this approach to be effective, the concept set $\widehat{\mathcal{C}}$ should be representative enough to cover the image classes $\mathcal{C}$ of interest. Instead of manually selecting the set $\widehat{\mathcal{C}}$, we leverage the zero-shot capability of foundation models where $\widehat{\mathcal{C}}$ is learned with only the visual information of the training set and *without seeing any class labels.*

Given the training dataset $\mathcal{D}^{\mathtt{T}} = \{\boldsymbol{X}_i\}_{i=1}^N$ and a vocabulary set $\mathcal{V} = \{v_j\}_{j=1}^{|\mathcal{V}|}$, we first measure the pairwise similarity score S_{ij} between $\boldsymbol{f}_i^{\mathtt{f}}$ and each vocabulary v_j, *i.e.*, $S_{ij} = \langle \boldsymbol{f}_i^{\mathtt{f}}, \mathtt{E}^{\mathtt{T}}(v_j)\rangle$. We then conduct a majority vote to

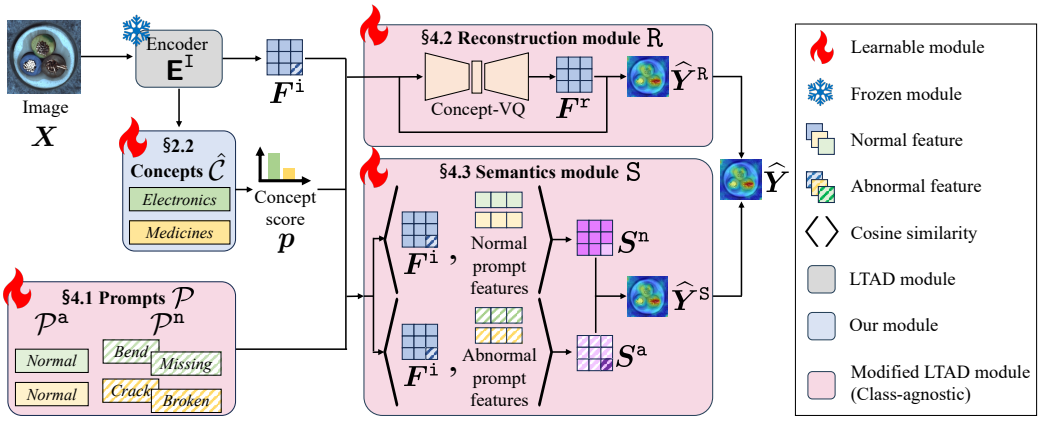


Figure 2. **Proposed class-agnostic pipeline.** We construct $\widehat{\mathcal{C}}$, and the correspondent normal prompts $\mathcal{P}^{\mathtt{n}}$ and abnormal prompts $\mathcal{P}^{\mathtt{a}}$. The concept score $\boldsymbol{p}$ is assigned to each image $\boldsymbol{X}$ by computing the similarity between $\boldsymbol{X}$ and each $\widehat{c} \in \widehat{\mathcal{C}}$. It then controls the soft switching mechanism in our class-agnostic reconstruction module $\mathtt{R}$ and semantics module $\mathtt{S}$. $\mathtt{R}$ reconstructs $\boldsymbol{F}^{\mathtt{i}}$ into $\boldsymbol{F}^{\mathtt{r}}$ through Concept VQ and output $\widehat{\boldsymbol{Y}}^{\mathtt{R}}$ by measuring the dissimilarity between $\boldsymbol{F}^{\mathtt{i}}$ and $\boldsymbol{F}^{\mathtt{r}}$. $\mathtt{S}$ compares the similarity map $\boldsymbol{S}^{\mathtt{n}}$ of $\boldsymbol{F}^{\mathtt{i}}$ and normal prompt features and the similarity map $\boldsymbol{S}^{\mathtt{a}}$ of $\boldsymbol{F}^{\mathtt{i}}$ and abnormal prompt features to output $\widehat{\boldsymbol{Y}}^{\mathtt{S}}$. The final prediction $\widehat{\boldsymbol{Y}}$ is aggregated from $\widehat{\boldsymbol{Y}}^{\mathtt{R}}$ and $\widehat{\boldsymbol{Y}}^{\mathtt{S}}$.

pick the top-$\widehat{K}$ vocabularies with the overall highest similarity scores across all images. This voting result is used to initialize $\widehat{\mathcal{C}}$. Specifically, we create embeddings $\boldsymbol{t}_{\widehat{c}} = \mathtt{E}^{\mathtt{T}}(\widehat{c})$ for all $\widehat{c} \in \widehat{\mathcal{C}}$ and make them learnable so that we can find more representative concept features during training.

At test time, a soft label, *i.e.*, concept score $\boldsymbol{p} \in [0,1]^{\widehat{K}}$, is predicted for each image $\boldsymbol{X}$. We compute $\boldsymbol{p}$ by measuring its similarity to all concepts $\widehat{c} \in \widehat{\mathcal{C}}$, *i.e.*,

$$\boldsymbol{p} = \mathtt{SoftMax}(\{\langle \boldsymbol{f}^{\mathtt{f}}, \boldsymbol{t}_{\widehat{c}} \rangle\}_{\widehat{c} \in \widehat{\mathcal{C}}}). \tag{1}$$

As shown in Fig. 1, we incorporate a soft switching mechanism and the final output is a composition of each $\boldsymbol{M}_{\widehat{c}}$ weighted by $\boldsymbol{p}$, *i.e.*, $\sum_{\widehat{c}=1}^{\widehat{K}} p_{\widehat{c}} \boldsymbol{M}_{\widehat{c}}$. We will now describe the high-level idea of how to incorporate soft switching into the AD pipeline.

2.3. Proposed class-agnostic AD pipeline

Our class-agnostic model is based on the general two-branch pipeline of LTAD [34] as it is the SOTA method, however, we note that the idea of soft switching can be applied to other class-aware models as well. Leveraging the previously acquired $\boldsymbol{p}$, our soft switching mechanism can learn to composite outputs resulting from any multiple concepts or class-specific submodules.

As shown in Fig. 2, we deploy a similar overall pipeline as LTAD but with several key differences: (a) we introduce $\widehat{C}$ as the cornerstone of our class-agnostic framework; (b) we replace the LTAD's class-aware modules with our class-agnostic LTOAD design; (c) for the model architecture in $\mathtt{R}$, we use our concept-based vector quantization VAE (Concept VQ); (d) we implement a comprehensive prompt learning. We provide the details of each module in Sec. 4.

3. Long-tailed online anomaly detection

3.1. Task formulation

Given a model $\mathtt{F}_{\theta_0}$ where θ_0 is the parameters trained offline on an LTAD dataset, the goal of LTOAD is to update the parameters θ_t in an online manner to improve the performance on a data stream $\widetilde{\boldsymbol{X}}_{\leq t} = \left[\widetilde{\boldsymbol{X}}_1, \cdots, \widetilde{\boldsymbol{X}}_t\right]$ where each image $\widetilde{\boldsymbol{X}}$ comes sequentially. Formally, we focus on improving the accuracy of $\mathtt{F}_{\theta_t}(\widetilde{\boldsymbol{X}}_{\leq t}) = \widehat{\boldsymbol{Y}}_t$, where $\widehat{\boldsymbol{Y}}_t$ is the prediction at t. Note that $\widetilde{\boldsymbol{X}}_{\leq t}$ is an ordered list, *i.e.*, LTOAD is evaluated sequentially.

Benchmark. We evaluate LTOAD by considering the any-Δ inference setting where the model is updated on small batches of data samples of size Δ. We split the data into two sets: **(a)** an online training set $\mathcal{D}^{\mathtt{0}} = \widetilde{\boldsymbol{X}}_{\leq N^{\mathtt{0}}}$ where $N^{\mathtt{0}}$ denotes the number of online training samples and **(b)** an evaluation set $\mathcal{D}^{\mathtt{E}}$. Given the training set $\mathcal{D}^{\mathtt{0}}$ and the model parameters $\theta_{t+1} \leftarrow \mathcal{A}(\theta_t; \mathcal{D}^{\mathtt{0}}_t)$ is updated based on the t-th batch of samples $\mathcal{D}^{\mathtt{0}}_t = \widetilde{\boldsymbol{X}}_{(t-1)\Delta:t\Delta}$ using an online learning algorithm $\mathcal{A}$. The model is then evaluated on $\mathcal{D}^{\mathtt{E}}$ at every step t.

Following standard metrics [34], we use AUROC to evaluate the performance of detection (Det.) and segmentation (Seg.). We report the average, over $\mathcal{C}^{\mathtt{head}}$, $\mathcal{C}^{\mathtt{tail}}$, and $\mathcal{C}$, at each intermediate steps $t' \in [1, T-1]$ in the data stream.

Dataset setup. As the data comes in a stream, a model's online learning performance is highly related to the ordering data in $\mathcal{D}^{\mathtt{0}}$. To study the effect, we sequentially split $\mathcal{D}^{\mathtt{0}}$ into *sessions* which corresponds to a sublist of the dataset. In each session, the distribution of the data is similar.

We consider two settings, *disjoint* and *blurry*, based on whether the class subset in each session overlaps or not. In a *disjoint* setting, the class subset in each session does not

Config.	Definition
B	a *blurry* stream where the whole stream is ordered randomly
B-HF	a *blurry* stream where $\mathcal{C}^{\texttt{head}}$ are more likely to appear first
B-TF	a *blurry* stream where $\mathcal{C}^{\texttt{tail}}$ are more likely to appear first
D2-HF	a *disjoint* stream with 2 sessions where all $\mathcal{C}^{\texttt{head}}$ are in the first session
D2-TF	a *disjoint* stream with 2 sessions where all $\mathcal{C}^{\texttt{tail}}$ are in the first session
D5-HF	a *disjoint* stream with 5 sessions where $\mathcal{C}^{\texttt{head}}$ concentrate in the first few sessions
D5-TF	a *disjoint* stream with 5 sessions where $\mathcal{C}^{\texttt{tail}}$ concentrate in the first few sessions
D5-M	a *disjoint* stream with 5 sessions where $\mathcal{C}^{\texttt{head}}$ and $\mathcal{C}^{\texttt{tail}}$ are mixed in each session

Table 1. **Configurations for $\mathcal{D}^{\texttt{0}}$.** We define 8 configurations with combinations of different session type $\in \{blurry, disjoint\}$ and ordering type $\in \{head\text{-}first,\ head\text{-}first,\ else\}$.

overlap with that of any other sessions, but in a *blurry* setting, the class subset of each session can overlap.

Next, data order is also important in the performance due to overfitting and forgetting, *e.g.*, if $\mathcal{D}^{\texttt{0}}$ has more $\mathcal{C}^{\texttt{tail}}$ at the beginning, the performance of $\mathcal{C}^{\texttt{head}}$ might drop first due to overfitting while the performance of $\mathcal{C}^{\texttt{tail}}$ plateaus at the end of the stream due to forgetting. Therefore, we also consider whether $\mathcal{C}^{\texttt{head}}$ or $\mathcal{C}^{\texttt{tail}}$ are more likely to appear first in $\mathcal{D}^{\texttt{0}}$ to study the effect of forgetting and overfitting. Overall, we design 8 settings for $\mathcal{D}^{\texttt{0}}$ with different session and ordering types combinations, as defined in Tab. 1. Please refer to our Appendix A2 for more details.

3.2. Anomaly adaptive online learning

We propose an Anomalous Adaptive learning algorithm $\mathcal{A}^{\texttt{AA}}$ to tackle the issues of overfitting and forgetting for LTOAD. As LTOAD is an unsupervised one-class classification task, a naive online algorithm $\mathcal{A}^{\texttt{N}}$ is to directly use the same loss function as offline learning. Note that $\mathcal{A}^{\texttt{N}}$ is not ideal as it does not leverage the unlabeled abnormal image in the online learning data.

To take advantage of this difference, our anomaly adaptive online learning algorithm $\mathcal{A}^{\texttt{AA}}$ computes a pseudo abnormal map $\widehat{\boldsymbol{M}} = \mathcal{T}(\widehat{\boldsymbol{Y}})$ acquired by thresholding $\widehat{\boldsymbol{Y}}$ with a function $\mathcal{T}$ to adjust the behavior of the original losses. The algorithm is summarized in Alg. 1 where we denote these adjustments as a mask-dependent loss function $\widetilde{\mathcal{L}}(\widehat{\boldsymbol{M}})$. As abnormal images in the online training set are unavailable in offline training, we value these samples more in the loss calculation and assign a larger weight β to images with higher $\widehat{y}$. Finally, we use the exponential moving average (EMA) for more stable updates. Please see the Appendix A1.5 for details.

4. Model details

We now provide the description for each of the modules as overviewed in Fig. 2. Additional implementation details are provided in the Appendix A1.

4.1. Prompt learning

Based on the acquired $\widehat{\mathcal{C}}$, we construct normal and abnormal prompt sets $\mathcal{P}^{\texttt{n}}$ and $\mathcal{P}^{\texttt{a}}$ for LTOAD to learn the textual

Algorithm 1 Anomaly Adaptive ($\mathcal{A}^{\texttt{AA}}$) Online learning

Input: offline-trained model with its parameters $\texttt{F}_{\theta_0}$, online training data stream $\mathcal{D}^{\texttt{0}}$, batch size Δ, threshold function $\mathcal{T}$, mask-dependent loss function $\widetilde{\mathcal{L}}$, reduction operation r, hyper-parameters $\tau,\ \beta,\ \gamma$

Output: $\Theta = [\theta_1, \cdots, \theta_T]$, where $T = |\mathcal{D}^{\texttt{0}}|/\Delta$

Initialize $\Theta = [\,]$, $\widetilde{\theta}_0 = \theta_0$
for $t = 1$ to T **do**
  Acquire batch $\mathcal{D}_t^{\texttt{0}} = \left[\widetilde{\boldsymbol{X}}_i\right]_{i=(t-1)\Delta,\cdots,t\Delta}$
  for $j = 1$ to Δ **do**
    Forward pass $\widehat{\boldsymbol{Y}}_j = \texttt{F}_{\widetilde{\theta}_{t-1}}(\mathcal{D}_t^{\texttt{0}}[j])$
    Compute pseudo abnormal label $\widehat{\boldsymbol{M}}_j = \mathcal{T}(\widehat{\boldsymbol{Y}}_j)$
    Compute mask-dependent loss $\widetilde{\mathcal{L}}_j = \widetilde{\mathcal{L}}(\widehat{\boldsymbol{M}}_j)$
    Compute gradient weight $\lambda_j \leftarrow \beta$ if $r(\widehat{\boldsymbol{Y}}_j) \geq \tau$ else 1
  end for
  $\widetilde{\theta}_t \leftarrow \widetilde{\theta}_{t-1} + \frac{1}{\Delta}\sum_j \lambda_j \nabla \widetilde{\mathcal{L}}_j$
  Update with EMA $\theta_t \leftarrow \gamma\theta_{t-1} + (1-\gamma)\widetilde{\theta}_t$
  Θ.append(θ_t)
end for
return Θ

semantics. $\mathcal{P}^{\texttt{n}}$ contains $\widehat{K}$ prompts where each $P^{\texttt{n}}_{\widehat{c}} \in \mathcal{P}^{\texttt{n}}$ is initialized with the template of "a normal $\widehat{c}$." In contrast, $\mathcal{P}^{\texttt{a}}$ is initialized by asking a conversational AI [13, 64] "For $\widehat{c}$, what are the 5 anomalies we are likely to observe?" for each $\widehat{c}$. Each normal prompt feature is defined as $\boldsymbol{n}_{\widehat{c}} = \texttt{E}^{\texttt{T}}(P^{\texttt{n}}_{\widehat{c}})$ while each abnormal prompt feature is defined as $\boldsymbol{a}_{\widehat{c},i} = \texttt{E}^{\texttt{T}}(P^{\texttt{a}}_{\widehat{c},i})$ for $i \in [1,\ 5]$. In addition to the learnable $\widehat{\mathcal{C}}$, $\boldsymbol{n}_{\widehat{c}}$ and $\boldsymbol{a}_{\widehat{c},i}$ are also learnable so that LTOAD can find more accurate abnormal semantics for each concept. In general, we learn $\boldsymbol{t}_{\widehat{c}}$, $\boldsymbol{a}_{\widehat{c}}$, and multiple $\boldsymbol{n}_{\widehat{c}}$ for each concept.

4.2. Reconstruction module

The goal of the reconstruction module $\texttt{R}$ is to reconstruct the features $\boldsymbol{F}^{\texttt{i}}$ of normal images for extracting cues about whether an image is anomalous. Since $\texttt{R}$ is trained only on the representation of normal features, it would be unable to accurately reconstruct abnormal features. Therefore, a discrepancy between the input and reconstructed features provides a useful cue that the input is abnormal. In more detail, given an input feature $\boldsymbol{F}^{\texttt{i}}$, $\texttt{R}$ reconstructs it to $\boldsymbol{F}^{\texttt{r}}$ and output an abnormal map prediction:

$$\widehat{\boldsymbol{Y}}^{\texttt{R}} = \frac{1}{2}\left(\boldsymbol{1} - \langle \boldsymbol{F}^{\texttt{i}}, \boldsymbol{F}^{\texttt{r}} \rangle\right) \tag{2}$$

by measuring the discrepancy between $\boldsymbol{F}^{\texttt{i}}$ and $\boldsymbol{F}^{\texttt{r}}$ via cosine similarity. We now introduce the background of HVQ [57] before we describe the module's architecture and concept codebooks used to create $\boldsymbol{F}^{\texttt{r}}$.

Background. Lu et al. [57] proposes hierarchical vector-quantization (HVQ) to construct multiple expertise modules

for each class to solve unified AD. A hard switching mechanism is applied to choose the correspondent modules for each class. They also introduce a hierarchical architecture to connect high-level and low-level feature maps. Given multiscale feature maps $\{\boldsymbol{Z}_l\}_{l=1,\cdots,L}$, where L is the number of layers, they build a hierarchical class-specific module $\mathtt{Q}_{l,c}$ for each $c \in \mathcal{C}$. During training, each $\mathtt{Q}_{l,c}$ only sees $\boldsymbol{Z}_l$ that comes from the images of class c. The output of each $\mathtt{Q}_{l,c}$ is denoted as $\boldsymbol{Q}_l$. Each $\boldsymbol{Q}_l$ is conditioned on $\boldsymbol{Z}_l$, and the output of the previous module, *i.e.* $\boldsymbol{Q}_{l-1}$. Specifically,

$$\boldsymbol{Q}_l = \begin{cases} \mathtt{Q}_{l,c}(\boldsymbol{Z}_l), & \text{if } l = 1 \\ \mathtt{Q}_{l,c}(\boldsymbol{Z}_l \oplus \boldsymbol{Q}_{l-1}), & \text{otherwise.} \end{cases} \tag{3}$$

Here, $\oplus$ denotes concatenation on the channel dimension. The outputs $\{\boldsymbol{Q}_l\}_{l=1,\cdots,L}$ are then fed into their proposed hierarchical decoder [57] for reconstruction.

The class-specific hierarchical modules $\mathtt{Q}_{l,c}$ are quantization modules [20, 59, 82] that learn to quantify visual feature representations. For readability, we remove the subscripts of $\mathtt{Q}_{l,c}$. Each $\mathtt{Q}$ stores a learnable codebook $\boldsymbol{B}$. The codebook contains M codes of D dimension, *i.e.* $\boldsymbol{B} \in \mathbb{R}^{M \times D}$. Given a 2D input feature $\boldsymbol{Z}$, $\mathtt{Q}$ queries its codebook and finds the nearest prototypes $\boldsymbol{Q}$. For each pixel location $(h,\ w)$, they find $\boldsymbol{Q}[h,\ w] = \boldsymbol{b}_{m^*}$, where $m^* \triangleq \arg\min_m \|\boldsymbol{Z}[h,\ w] - \boldsymbol{B}[m]\|_2^2$.

Module architecture and concept codebooks. Given an input $\boldsymbol{F}^{\mathtt{i}}$, we first project it onto a visual-text shared space through a layer-wise projection module $\mathtt{E}^{\mathtt{p}}$ [34], *i.e.*, $\boldsymbol{F}^{\mathtt{p}} = \mathtt{E}^{\mathtt{p}}(\boldsymbol{F}^{\mathtt{i}})$. For feature reconstruction, we use a VQ-autoencoder dubbed "Concept-VQ," modified from class-specific HVQ [57] for our concept-specific objective. We also use a prompt-conditioned generator $\mathtt{G}$ for data augmentation to generate pseudo-normal features $\boldsymbol{F}^{\mathtt{n}}$ and pseudo-abnormal features $\boldsymbol{F}^{\mathtt{a}}$. These pseudo features provide additional guidance in training $\mathtt{R}$. Finally, we use the soft label $\boldsymbol{p}$ as a weighting mechanism for our concept-specific designs in Concept-VQ and $\mathtt{G}$.

Our Concept-VQ contains an encoder $\mathtt{E}^{\mathtt{R}}$, a decoder $\mathtt{D}^{\mathtt{R}}$, and hierarchical quantization modules between them. For each $l \in [1, \cdots, L]$ and each $\widehat{c} \in \widehat{\mathcal{C}}$, we construct a concept-specific quantization module $\mathtt{Q}_{l,\widehat{c}}$. To capture the text-visual representation of $\mathcal{D}^{\mathtt{T}}$, our concept codebook $\boldsymbol{B}_{l,\widehat{c}}$ in each $\mathtt{Q}_{l,\widehat{c}}$ is initialize $\boldsymbol{B}_{l,\widehat{c}}$ by sampling M codes around its correspondent $\boldsymbol{t}_{\widehat{c}}$.

Prompt-conditioned data augmentation. During training, prior works [21, 34, 57, 89] create pseudo-abnormal features for data augmentation by adding noise to the input features, *i.e.*, *feature jittering*. These prior works ignore the semantics of the $\boldsymbol{F}^{\mathtt{p}}$. Instead, we build a generator $\mathtt{G}$ that synthesizes pseudo features conditioned on prompt features $\boldsymbol{a}$ and $\boldsymbol{n}$. We generate pseudo-normal features $\boldsymbol{F}^{\mathtt{n}}$ conditioned on $\boldsymbol{F}^{\mathtt{p}}$ and $\boldsymbol{n}$ while we generate pseudo-abnormal features $\boldsymbol{F}^{\mathtt{a}}$ conditioned on $\boldsymbol{F}^{\mathtt{p}}$ and $\boldsymbol{a}$.

4.3. Semantics module

The semantics module $\mathtt{S}$ leverages the cross-modality knowledge of abnormality in $\mathtt{E}^{\mathtt{I}}$ and $\mathtt{E}^{\mathtt{T}}$ to extract cues on whether an image is anomalous. It learns the semantic similarities or dissimilarities between visual features and normal texts, *e.g.*, *"normal,"* versus abnormal texts, *e.g.*, *"broken"* by comparing the aggregated normal similarity scores $\boldsymbol{S}^{\mathtt{n}}$ and abnormal scores $\boldsymbol{S}^{\mathtt{a}}$. The higher $\boldsymbol{S}^{\mathtt{a}} - \boldsymbol{S}^{\mathtt{n}}$ is, the more likely it is abnormal. Given the projected feature $\boldsymbol{F}^{\mathtt{p}}$, we denote the similarity difference as a functionality of $\mathtt{S}$, *i.e.*, $\mathtt{S}(\boldsymbol{F}^{\mathtt{p}}) = \boldsymbol{S}^{\mathtt{a}} - \boldsymbol{S}^{\mathtt{n}}$. Then, the final prediction $\widehat{\boldsymbol{Y}}^{\mathtt{S}} \triangleq (\mathtt{S}(\boldsymbol{F}^{\mathtt{p}}) + \mathbf{2})/4$, where we scale to the range of $[0,\ 1]$.

Module architecture. Given $\boldsymbol{F}^{\mathtt{p}}$, we first compute the similarity map $\boldsymbol{S}^{\mathtt{n}}_{\widehat{c}}$ between it and the concept-specific normal prompt features, *i.e.*, $\boldsymbol{S}^{\mathtt{n}}_{\widehat{c}} = \langle \boldsymbol{F}^{\mathtt{p}},\ \boldsymbol{n}_{\widehat{c}} \rangle$. The overall normal similarity map $\boldsymbol{S}^{\mathtt{n}}$ is then $\sum_{\widehat{c}} p_{\widehat{c}} \boldsymbol{S}^{\mathtt{n}}_{\widehat{c}}$. Similarly, we also compute the overall abnormal similarity map $\boldsymbol{S}^{\mathtt{a}}$ for $\boldsymbol{F}^{\mathtt{p}}$.

For segmentation, the final prediction $\widehat{\boldsymbol{Y}}$ is the elementwise harmonic mean [40, 51] of the two intermediate predictions, *i.e.*,

$$\widehat{\boldsymbol{Y}} = \left(\alpha(\widehat{\boldsymbol{Y}}^{\mathtt{R}})^{-1} + (1-\alpha)(\widehat{\boldsymbol{Y}}^{\mathtt{S}})^{-1}\right)^{-1}, \tag{4}$$

where α controls the weight between them. For detection, we apply an `AvgPool2d` with a kernel size of 16 on $\widehat{\boldsymbol{Y}}$ before we reduce the map to its maximum value as $\widehat{y}$. We denote this reduction operation as $\widehat{y} = r(\widehat{\boldsymbol{Y}})$.

5. Experiments

We first study the offline LTAD setting as an offline model parameter θ_0 is the starting point for an online method. Importantly, we show that class-agnostic models are effective in offline LTAD without degrading performance. Finally, we conduct experiments on LTOAD with our $\mathcal{A}^{\mathtt{AA}}$ in the online setting and evaluate cross-dataset generalizability. Additional experiment details are in the Appendix A3.

5.1. Offline long-tailed settings

Experiment setup. Following the LTAD benchmark [34], we report on MVTec [5] and VisA [103]. Each dataset has multiple long-tailed settings dependent on the degree of imbalance between $\mathcal{C}^{\mathtt{head}}$ and $\mathcal{C}^{\mathtt{tail}}$. The imbalance type $\in \{\textit{exp},\ \textit{step}\}$ describes whether the numbers of classes decay exponentially or drop like a step function; while the imbalance factor $\in \{\textit{100},\ \textit{200}\}$ measures the ratio between the number of the most populated class to the least one. A larger imbalance factor means a more challenging setting. We follow exactly the same LTAD training and testing settings unless otherwise specified. Additional results on DAGM [85] and Uni-Medical [4, 95] are provided in the Appendix A3.

Method	CA	*exp100*		*exp200*		*step100*		*step200*	
		Det.	Seg.	Det.	Seg.	Det.	Seg.	Det.	Seg.
RegAD [36]	✗	82.43	95.20	N/A	N/A	81.54	95.10	N/A	N/A
AnomalyGPT [25]	✗	87.44	89.68	85.80	90.15	85.95	89.28	82.47	89.45
LTAD [34]	✗	88.86	94.46	86.05	94.18	87.36	93.83	85.60	92.12
HVQ [57]	✗	87.43	**95.25**	85.34	94.79	85.39	94.17	83.00	92.61
LTOAD*	✗	93.12	95.01	91.88	94.36	92.02	94.72	87.95	93.34
UniAD [89]	✓	87.70	93.95	86.21	93.26	83.37	91.47	81.32	92.29
MoEAD [62]	✓	84.73	94.34	83.23	93.96	84.40	93.76	83.13	92.76
LTOAD	✓	**93.42**	95.21	**92.02**	**94.94**	**92.33**	**95.11**	**88.62**	**94.00**

Table 2. **Comparison (↑) on offline MVTec** in image-level AUROC for anomaly detection (Det.) and pixel-level AUROC for anomaly segmentation (Seg.). The column CA (class-agnostic) indicates whether a method requires class names or the number of classes during training or not (require: ✗; not require: ✓). RegAD needs at least 2 images per class and thus is not applicable on *exp200* and *step200*. We mark the best and second best performances in **bold** and underline.

Method	CA	*exp100*		*exp200*		*step100*		*step200*	
		Det.	Seg.	Det.	Seg.	Det.	Seg.	Det.	Seg.
RegAD [36]	✗	71.36	94.40	71.36	94.40	71.80	94.99	71.65	94.52
AnomalyGPT [25]	✗	70.34	80.32	69.78	79.48	71.98	82.30	69.78	81.97
LTAD [34]	✗	80.00	95.56	80.21	95.36	84.80	96.57	84.03	96.27
HVQ [57]	✗	78.63	96.18	78.03	95.32	76.68	95.63	75.52	95.50
LTOAD*	✗	84.89	96.42	84.84	96.44	87.71	97.07	86.52	96.89
UniAD [89]	✓	77.31	95.03	76.87	94.80	78.83	96.04	77.64	95.66
MoEAD [62]	✓	73.89	94.34	73.58	93.98	74.12	94.37	76.77	94.82
LTOAD	✓	**85.26**	**96.91**	**85.24**	**96.97**	**88.32**	**97.54**	**87.04**	**97.23**

Table 3. **Comparison (↑) on offline VisA**. Format follows Tab. 2.

Baselines & implementation details. We compare with RegAD [36], UniAD [89], and LTAD [34]. We benchmark the SOTA MoEAD [62] and HVQ [57] to highlight the gain of our concept-specific quantization modules to their class-specific ones. Only UniAD, MoEAD, and LTOAD are class-agnostic and do not need a class label at the test time. We provide a class-aware version of our model, LTOAD*.

Quantitative & qualitative comparison. Tab. 2 and 3 show the comparisons of LTOAD with the baselines on MVTec and VisA. We observe that LTOAD outperforms the SOTA on most settings while using no class information. LTOAD achieves a larger gain over Det. On average across all settings, LTOAD achieves +4.63% / +0.58% gain versus the second best baseline excluding LTOAD* on MVTec Det. / Seg. As for VisA, LTOAD achieves +4.30% / +1.29% gain.

Next, comparing LTAD and LTOAD*, we find that being class-agnostic is beneficial. We hypothesize that this is because the classes can now share knowledge through the soft labels. Finally, we show qualitative results in Fig. 3. We observe that LTOAD predictions are closer to ground truth than HVQ and LTAD.

5.2. Online long-tailed benchmark

Experiment setup. For the online setting, we report on the same datasets as the offline benchmark [34], *i.e.*, MVTec, VisA, and DAGM. We split the testing set into $\mathcal{D}^{\mathtt{O}}$ and $\mathcal{D}^{\mathtt{E}}$ with a 3:7 ratio. For each dataset, we create the 8 settings defined in Tab. 1.

For baseline comparisons, we extend HVQ and LTAD to have online capability by applying our naive online learning algorithm $\mathcal{A}^{\mathtt{N}}$ since they are the best offline models on MVTec. Note that both methods are not class-agnostic and needs class labels as extra information in online learning, which is not available/necessary for our approach.

Quantitative comparison. We report the online Seg. performance at final time step $t = T$ on MVTec *exp100* in Tab. 4. The results on other settings/datasets are in the

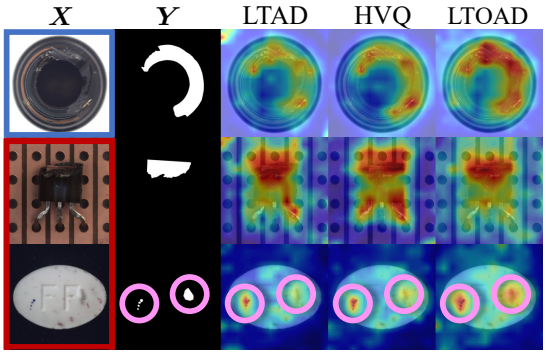


Figure 3. **Qualitative comparison** among LTAD [34], HVQ [57], and LTOAD on MVTec offline LTAD [34] *exp100* benchmark. Inputs from $\mathcal{C}^{\mathtt{head}}$ / $\mathcal{C}^{\mathtt{tail}}$ are outlined in blue / red. Smaller ground truth anomaly masks are circled in pink.

Method	CA	Online	*B*	*B-HF*	*B-TF*	*D2-HF*	*D2-TF*	*D5-HF*	*D5-TF*	*D5-M*	Avg.
LTAD [34]	✗	✗	93.46	93.46	93.46	93.46	93.46	93.46	93.46	93.46	93.46
LTAD [34]	✗	$\mathcal{A}^{\mathtt{N}}$	94.12	93.80	94.07	93.70	94.08	93.49	93.84	93.65	93.84
HVQ [57]	✗	✗	95.02	95.02	95.02	95.02	95.02	95.02	95.02	95.02	95.02
HVQ [57]	✗	$\mathcal{A}^{\mathtt{N}}$	**95.44**	**95.33**	95.28	95.22	**95.29**	95.03	95.21	95.05	95.23
LTOAD	✓	✗	95.08	95.08	95.08	95.08	95.08	95.08	95.08	95.08	95.08
LTOAD	✓	$\mathcal{A}^{\mathtt{N}}$	95.41	95.22	**95.32**	**95.23**	95.04	94.91	94.72	94.79	95.08
LTOAD	✓	$\mathcal{A}^{\mathtt{AA}}$	95.36	95.30	**95.32**	**95.23**	95.28	**95.26**	**95.25**	**95.21**	**95.28**

Table 4. **Comparison (↑) on online MVTec** in pixel-level AUROC for anomaly segmentation across 8 configurations of $\mathcal{D}^{\mathtt{O}}$. All models are pre-trained on MVTec *exp100* [34].

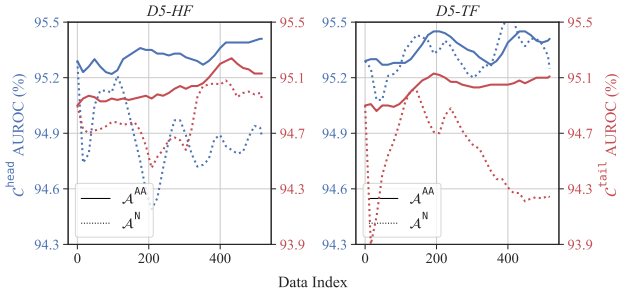


Figure 4. **Performance curve** of $\mathcal{A}^{\mathtt{N}}$ and $\mathcal{A}^{\mathtt{AA}}$ in pixel-level AUROC on $\mathcal{C}^{\mathtt{head}}$ and $\mathcal{C}^{\mathtt{tail}}$. They are offline-trained on *exp100* [34] and online-trained on *D5-HF* or *D5-TF*.

Method	Online	MVTec		VisA		DAGM	
		→ VisA	→ DAGM	→ MVTec	→ DAGM	→ MVTec	→ VisA
LTOAD	✗	84.38	91.27	82.73	79.16	79.51	81.13
LTOAD	✓	**92.37**	**96.87**	**89.84**	**96.79**	**88.37**	**92.41**

Table 5. **Cross-dataset** anomaly segmentation in pixel-level AUROC. All models are pre-trained on their *exp100* [34].

Set ID	Domain	Size	Elements
$\mathcal{C}$	N/A	15	zipper, pill, capsule, grid, transistor, carpet, metal nut, wood, leather, screw, tile, cable, toothbrush, hazelnut, bottle*
Random	Out	10	refinery, finger, vault, exam, salad, sailor, dove, dealer, well, humanoid
Small	In	3	semiconductor, zipper, beech
LTOAD $\widehat{\mathcal{C}}$	In	10	semiconductor, zipper, beech, microscopy*, antibiotics, medicines, circuit , mahogany, hardwood, walnut

Table 6. **Classes vs. concepts on MVTec.** Words with similar semantics have the same color, *e.g.*, *wood*-related words are colored in brown and *electronics*-related words are colored in purple.

Appendix A4. Overall, $\mathcal{A}^{\mathtt{N}}$ performs well in the simplest setting (*B*), *but degrades* on more difficult settings, *e.g.*, *D2-TF, D5-HF*, *etc*. On average, LTOAD with $\mathcal{A}^{\mathtt{AA}}$ performs the best even without class labels as extra information. We plot LTOAD's performance curve of $\mathtt{F}_{\theta_t}$ over t with Δ=16 under the two most difficult settings (*D5-HF* and *D5-TF*) in Fig. 4, where $\mathcal{A}^{\mathtt{N}}$ falls off on $\mathcal{C}^{\mathtt{head}}$/$\mathcal{C}^{\mathtt{head}}$ under *D5-HF*/*D5-TF* while $\mathcal{A}^{\mathtt{AA}}$ improves steadily. Empirically, LTOAD is also more efficient than LTAD because $\widehat{K} < K$.

Cross-dataset evaluation. Tab. 5 shows that LTOAD can adapt to unseen domains even if it is not designed for cross-dataset evaluation. For each $\mathcal{D}_1 \rightarrow \mathcal{D}_2$ experiment, we pre-train LTOAD on *exp100* of $\mathcal{D}_1$ then evaluated on *B* of $\mathcal{D}_2$ without and with online learning. LTOAD with online learning outperforms the one without it.

5.3. Additional analysis

Analysis on initialization of $\widehat{\mathcal{C}}$. We now analyze the effect of our concept learning component. All experiments are done under MVTec *exp100* and *B*. We highlight the importance of our concept learning compared to two baselines: initializing with a *random* concept set or with a *small* one.

Tab. 6 shows the learned initialization of $\widehat{\mathcal{C}}$ versus the ground truth $\mathcal{C}$. We observe that our concept learning can find the $\widehat{\mathcal{C}}$ initialization which represents similar semantics in $\mathcal{C}$, *e.g.*, the *bottle** images (see the 1st image in Fig. 3) in MVTec are bottom-only views, which look like an *microscopy**.

Next, we experiment with the two baselines to generate initialization for $\widehat{\mathcal{C}}$. In *Random* of Tab. 6, instead of learning from $\mathcal{D}^{\mathtt{T}}$, *i.e.*, in-domain concepts, we randomly select 10 nouns in ALIGN's $\mathcal{V}$ unrelated to the industrial settings of MVTec, *i.e.*, out-domain concepts. In other words, the semantics of $\mathcal{C}$ and *Random* do not overlap. In *Small* of Tab. 6, instead of choosing $\widehat{K} = 10$, we select only the top-$\widehat{K} = 3$ vocabularies as $\widehat{\mathcal{C}}$. The semantics of *Small* partially covers the semantics of $\mathcal{C}$. Tab. 7 shows the quantitative results.

Exp.	$\widehat{\mathcal{C}}$	$\widehat{K}$	R	S	G	Detection			Segmentation		
						$\mathcal{C}^{\mathtt{tail}}$	$\mathcal{C}^{\mathtt{head}}$	$\mathcal{C}$	$\mathcal{C}^{\mathtt{tail}}$	$\mathcal{C}^{\mathtt{head}}$	$\mathcal{C}$
Random	O-D	10	✓	✓	✓	84.73	99.31	91.54	94.46	95.39	94.89
Small	I-D	3	✓	✓	✓	85.66	98.89	91.83	94.24	95.30	94.74
NoR	I-D	10	✗	✓	✗	53.84	56.25	54.97	60.73	63.46	62.00
NoS	I-D	10	✓	✗	✓	85.16	99.21	91.72	94.57	95.24	94.88
NoG	I-D	10	✓	✓	✗	85.63	**99.56**	92.13	94.08	95.62	94.80
LTOAD	I-D	10	✓	✓	✓	**88.14**	99.45	**93.72**	**94.83**	**95.64**	**95.36**

Table 7. **Analysis and ablation on LTOAD** using the same metrics as Tab. 4 under MVTec *exp100* [34] and *B*. We report the performances on $\mathcal{C}^{\mathtt{head}}$ and $\mathcal{C}^{\mathtt{tail}}$. I-D/O-D: in/out-domain.

Exp.	$\widehat{\mathcal{C}}$	$\mathcal{P}^{\mathtt{a}}$			Detection			Segmentation		
	L	L	M	CS	$\mathcal{C}^{\mathtt{tail}}$	$\mathcal{C}^{\mathtt{head}}$	$\mathcal{C}$	$\mathcal{C}^{\mathtt{tail}}$	$\mathcal{C}^{\mathtt{head}}$	$\mathcal{C}$
FixC	✗	✓	✓	✓	84.65	98.13	90.94	94.57	95.16	94.85
FixP	✓	✗	✓	✓	85.59	**99.45**	92.06	94.52	94.75	95.04
SingleP	✓	✓	✗	✓	85.41	99.37	91.92	94.64	95.44	95.02
ShareP	✓	✓	✓	✗	87.03	99.35	92.78	95.00	95.63	**95.39**
LTOAD	✓	✓	✓	✓	**88.14**	**99.45**	**93.72**	**94.83**	**95.64**	95.36

Table 8. **Ablations on concept $\widehat{\mathcal{C}}$ & abnormal prompt $\mathcal{P}^{\mathtt{a}}$ learning** under MVTec *exp100* [34] on the same metrics as Tab. 7. Acronyms: L: learned; M: multiple; CS: concept-specific.

We report the performance of *Random* and *Small* in Tab. 7. Even with an out-domain initialization, compared to HVQ in Tab. 2, which uses $\mathcal{C}$ for their class-specific quantization modules, *Random* still performs competitively. Specifically, we observe a +4.11% gain in Det. versus HVQ. On top of that, with in-domain $\widehat{\mathcal{C}}$ as a more aligned representation, LTOAD in Tab. 7 achieves the best image-level AUROC. In particular, we further improve it by +1.88%. We have a similar observation initializing with *Small*, which outperforms HVQ on Det. Having sufficiently large $\widehat{\mathcal{C}}$, LTOAD with $\widehat{K} = 10$ gives more gain.

Finally, although some $c \in \mathcal{C}$ are not presented in our $\widehat{\mathcal{C}}$, LTOAD still adapts and outperforms the baselines in Tab. 7. This suggests that LTOAD can work even with a foundation model whose $\mathcal{V}$ and $\mathcal{C}$ are disjoint. However, this is not necessarily true for prior class-aware methods as they implicitly assume that $\mathcal{C} \subseteq \mathcal{V}$.

Ablation studies on R, S, and G. In Tab. 7, we also ablate LTOAD components (see Sec. 4.2 and 4.3) to justify their design. All experiments are done under the MVTec *exp100* and *B* setting. *NoR* ablates R. We also remove G, $\widehat{\boldsymbol{Y}}^{\mathtt{R}}$ from Eq. 4 and related losses (see Appendix A1.1). In contrast, *NoS* ablates S. We also remove $\widehat{\boldsymbol{Y}}^{\mathtt{S}}$ and related

Unsupervised AD Method Conditions	Unsupervised AD Categories								
	$\mathcal{G}_0$	$\mathcal{G}_1$	$\mathcal{G}_2$	$\mathcal{G}_3$	$\mathcal{G}_4$	$\mathcal{G}_5$	$\mathcal{G}_6$	$\mathcal{G}_7$	LTOAD
unified model	✗	✓	✓	✗	✓	✓	✗	✓	✓
online learning capability	✗	✗	✗	✗	✗	✗	✓	✓	✓
incorporate VLM/LLM(s)	✗	✗	✗	✓	✓	✓	✗	✗	✓
can handle long-tailed data	✗	✗	✗	✗	✗	✓	✗	✗	✓
class-agnostic	✗	✓	✗	✗	✗	✗	✗	✓	✓

Table 9. **Characteristic comparison** between LTOAD and prior unsupervised AD works (*e.g.*, $\mathcal{G}_0$: [12, 14, 15, 19, 26, 27, 38, 48, 54, 59, 63, 70, 72, 73, 75, 80, 87, 90, 91, 93, 97], $\mathcal{G}_1$: [2, 11, 21, 30, 31, 33, 39, 46, 50, 62, 66, 76, 77, 87–89, 95, 99, 102], $\mathcal{G}_2$: [57], $\mathcal{G}_3$: [37, 51, 52], $\mathcal{G}_4$: [7, 9, 10, 25, 40, 49, 67, 94], $\mathcal{G}_5$: [34], $\mathcal{G}_6$: [22, 61, 84], $\mathcal{G}_7$: [55, 79]).

losses. We find from *NoR* that S alone is not an effective branch while *NoS* is far more competitive. However, when trained jointly (LTOAD) and compared to *NoS*, S helps R improve the Det. accuracy on $\mathcal{C}^{\texttt{tail}}$ notably. Specifically, S improves the image-level AUROC on $\mathcal{C}^{\texttt{tail}}$ by 2.98%.

NoG replaces our prompt-conditioned G with unconditional *feature jittering* [34, 57, 89] for data augmentation. With the guidance of $\mathcal{P}^{\texttt{n}}$ and $\mathcal{P}^{\texttt{a}}$, G creates high-quality pseudo-features especially beneficial for $\mathcal{C}^{\texttt{tail}}$. We observe a gain of $+2.51\%$ / $+0.75\%$ in Det. / Seg. on $\mathcal{C}^{\texttt{tail}}$.

Ablation studies on learnable concepts/prompts. Tab. 8 analyzes the effect of each attribute detailed in Tab. 10, where we demonstrate that our prompt learning design is comprehensive. All experiments are performed in the setting MVTec *exp100* and *B*. *FixC* freezes $\widehat{\mathcal{C}}$. *FixP* freezes $\mathcal{P}^{\texttt{a}}$. In *SingleP*, $\mathcal{P}^{\texttt{a}}$ stores only one abnormal prompt for each concept. In *ShareP*, all concepts share the same abnormal prompts. Each attribute contributes significantly to the Det. performance of $\mathcal{C}^{\texttt{tail}}$ and $\mathcal{C}$.

6. Related work

In Tab. 9, we group the literature of unsupervised AD based on their assumptions or target tasks. Notably, LTOAD (1) is a unified model, (2) has online learning capability, (3) uses VLM/LLM priors, (4) handles long-tailed training data distributions, and (5) is class-agnostic.

Unsupervised anomaly detection. $\mathcal{G}_0$ methods train an independent predictor per class. In contrast, UniAD [89] challenges to learn one unified predictor for all classes. More follow-up research falls into this group ($\mathcal{G}_1$). Inspired by prior work [56, 60, 71], MoEAD [62] uses mixture-of-experts. HVQ [57] uses class-specific quantization modules in their VQ-VAE [82] architecture, but HVQ also induces the constraint of requiring class information as inputs, making it not class-agnostic, *i.e.*, $\mathcal{G}_2$.

Foundation model based AD. With advancement in foundation models, recent works focused on zero-shot [7, 9, 24, 40, 50, 67, 100, 102] or few-shot [2, 25, 51–53, 101] unsupervised AD where the normal training images are limited (or none), they [7, 9, 25, 40, 51, 52, 67, 100] often turn themselves to the rich knowledge in foundation models [29, 41, 68, 81, 98] or conversational AI [64]. These methods can be grouped into either $\mathcal{G}_3$ or $\mathcal{G}_4$, depending on whether they are unified model. To query the language models precisely, $\mathcal{G}_3$ and $\mathcal{G}_4$ need the class names or labels as extra inputs and are thus not class-agnostic.

Prompt Design Conditions of Unsupervised AD Methods	Prompt Design Categories in AD					
	$\mathcal{G}_8$	$\mathcal{G}_9$	$\mathcal{G}_{10}$	$\mathcal{G}_{11}$	$\mathcal{G}_{12}$	LTOAD
learnable class/concept names	✓	✗	✗	✗	✗	✓
support >1 APs	✗	✗	✓	✓	✓	✓
learnable APs	✗	✓	✗	✗	✓	✓
class/concept specific APs	✗	✗	✗	✓	✓	✓

Table 10. **Characteristic comparison w.r.t. prompt design** between LTOAD and prior unsupervised AD works which use LLM(s) with prompts (*e.g.*, $\mathcal{G}_8$: [34, 67], $\mathcal{G}_9$: [10], $\mathcal{G}_{10}$: [9, 37, 40, 52, 101], $\mathcal{G}_{11}$: [7], $\mathcal{G}_{12}$: [51, 102]). AP: abnormal prompts.

Although normal images are limited in zero/few-shot AD, the distribution is even since the number of images per class is the same. Consequently, LTAD [34], *i.e.*, $\mathcal{G}_5$, explores the long-tailed setting where the training set is highly unevenly distributed. They rely on the pre-trained ALIGN [41] and learn the pseudo-class names. We highlight the differences between LTOAD prompt learning to related works in Tab. 10. In summary, LTOAD supports (a) learnable concept names, (b) learnable abnormal prompts (AP), (c) multiple APs, and (d) concept-specific APs.

Online learning AD. Online (or continual/incremental) learning [23, 44, 69, 83] investigates how the model learns continuously on a data stream. Recent works [22, 55, 61, 79, 84], *i.e.* $\mathcal{G}_6$ and $\mathcal{G}_7$, explore unsupervised online learning AD. Differently, we propose the task of long-tailed online anomaly detection to study the effect of head/tail classes.

7. Conclusion

We propose the benchmark of long-tailed online anomaly detection, studying the online performance of head and tail classes. We identify the shortcomings of class-aware methods and propose class-agnostic LTOAD. It learns to approximate the ground truth class set while using foundation models for our novel concept and abnormal prompt learning framework. We also propose $\mathcal{A}^{\texttt{AA}}$, a learning algorithm to adapt the online stream with potentially abnormal inputs. LTOAD outperforms the SOTA in most offline and online settings without additional class information. Although not specifically designed for domain generalization, LTOAD can adapt to unseen domains. We believe that LTOAD will advance the field toward a more realistic anomaly detection setting that is closer to real-world applications.

Acknowledgments. CAY and KCP were supported by Mitsubishi Electric Research Laboratories. CAY and RAY were in part supported by an NSF Award #2420724.

References

[1] Akshatha Arodi, Margaux Luck, Jean-Luc Bedwani, Aldo Zaimi, Ge Li, Nicolas Pouliot, Julien Beaudry, and Gaétan Marceau Caron. CableInspect-AD: An expert-annotated anomaly detection dataset. In *Proc. NeurIPS*, 2024. 1

[2] Yuhu Bai, Jiangning Zhang, Zhaofeng Chen, Yuhang Dong, Yunkang Cao, and Guanzhong Tian. Dual-path frequency discriminators for few-shot anomaly detection. *Knowledge-Based Systems*, 2024. 8

[3] Ujjwal Baid, Satyam Ghodasara, Suyash Mohan, Michel Bilello, Evan Calabrese, Errol Colak, Keyvan Farahani, Jayashree Kalpathy-Cramer, Felipe C Kitamura, Sarthak Pati, et al. The rsna-asnr-miccai brats 2021 benchmark on brain tumor segmentation and radiogenomic classification. *arXiv preprint arXiv:2107.02314*, 2021. 5

[4] Jinan Bao, Hanshi Sun, Hanqiu Deng, Yinsheng He, Zhaoxiang Zhang, and Xingyu Li. BMAD: Benchmarks for medical anomaly detection. In *Proc. CVPR*, 2024. 5, 1

[5] Paul Bergmann, Michael Fauser, David Sattlegger, and Carsten Steger. MVTec AD–a comprehensive real-world dataset for unsupervised anomaly detection. In *Proc. CVPR*, 2019. 1, 5, 3, 4

[6] Patrick Bilic, Patrick Christ, Hongwei Bran Li, Eugene Vorontsov, Avi Ben-Cohen, Georgios Kaissis, Adi Szeskin, Colin Jacobs, Gabriel Efrain Humpire Mamani, Gabriel Chartrand, et al. The liver tumor segmentation benchmark (lits). *Medical Image Analysis*, 2023. 5

[7] 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 preprint arXiv:2305.10724*, 2023. 1, 2, 8

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

[9] Yunkang Cao, Jiangning Zhang, Luca Frittoli, Yuqi Cheng, Weiming Shen, and Giacomo Boracchi. AdaCLIP: Adapting CLIP with hybrid learnable prompts for zero-shot anomaly detection. In *Proc. ECCV*, 2024. 1, 2, 8

[10] Pi-Wei Chen, Jerry Chun-Wei Lin, Jia Ji, Feng-Hao Yeh, Zih-Ching Chen, and Chao-Chun Chen. Human-free automated prompting for vision-language anomaly detection: Prompt optimization with meta-guiding prompt scheme. *arXiv preprint arXiv:2406.18197*, 2024. 1, 2, 8

[11] Qiyu Chen, Huiyuan Luo, Chengkan Lv, and Zhengtao Zhang. A unified anomaly synthesis strategy with gradient ascent for industrial anomaly detection and localization. In *Proc. ECCV*, 2024. 8

[12] Niv Cohen and Yedid Hoshen. Sub-image anomaly detection with deep pyramid correspondences. *arXiv preprint arXiv:2005.02357*, 2020. 8

[13] Microsoft Corporation. Microsoft Copilot, 2023. 4, 2

[14] Thomas Defard, Aleksandr Setkov, Angelique Loesch, and Romaric Audigier. PaDiM: a patch distribution modeling framework for anomaly detection and localization. In *Proc. ICPR*, 2021. 8

[15] Hanqiu Deng and Xingyu Li. Anomaly detection via reverse distillation from one-class embedding. In *Proc. CVPR*, 2022. 8

[16] Keval Doshi and Yasin Yilmaz. Continual learning for anomaly detection in surveillance videos. In *Proc. CVPR Workshop*, 2020. 1

[17] Hugging Face. Transformers: Align model documentation. https://huggingface.co/docs/transformers/model_doc/align. 2

[18] Yehonatan Fridman, Matan Rusanovsky, and Gal Oren. ChangeChip: A reference-based unsupervised change detection for PCB defect detection. In *Proc. PAINE*, 2021. 1

[19] Matic Fučka, Vitjan Zavrtanik, and Danijel Skočaj. TransFusion–a transparency-based diffusion model for anomaly detection. In *Proc. ECCV*, 2024. 8

[20] Hugo Gangloff, Minh-Tan Pham, Luc Courtrai, and Sébastien Lefèvre. Leveraging vector-quantized variational autoencoder inner metrics for anomaly detection. In *Proc. ICPR*, 2022. 1, 5

[21] Bin-Bin Gao. Learning to detect multi-class anomalies with just one normal image prompt. In *Proc. ECCV*, 2024. 5, 8

[22] Han Gao, Huiyuan Luo, Fei Shen, and Zhengtao Zhang. Towards total online unsupervised anomaly detection and localization in industrial vision. *arXiv preprint arXiv:2305.15652*, 2023. 1, 2, 8

[23] Yasir Ghunaim, Adel Bibi, Kumail Alhamoud, Motasem Alfarra, Hasan Abed Al Kader Hammoud, Ameya Prabhu, Philip HS Torr, and Bernard Ghanem. Real-time evaluation in online continual learning: A new hope. In *Proc. CVPR*, 2023. 8

[24] Zhaopeng Gu, Bingke Zhu, Guibo Zhu, Yingying Chen, Hao Li, Ming Tang, and Jinqiao Wang. Filo: Zero-shot anomaly detection by fine-grained description and high-quality localization. In *Proc. ACM MM*, 2024. 8

[25] Zhaopeng Gu, Bingke Zhu, Guibo Zhu, Yingying Chen, Ming Tang, and Jinqiao Wang. AnomalyGPT: Detecting industrial anomalies using large vision-language models. In *Proc. AAAI*, 2024. 2, 6, 8, 5, 7, 9, 10, 11, 12, 13, 14

[26] Guan Gui, Bin-Bin Gao, Jun Liu, Chengjie Wang, and Yunsheng Wu. Few-shot anomaly-driven generation for anomaly classification and segmentation. In *Proc. ECCV*, 2024. 8

[27] Jia Guo, Haonan Han, Shuai Lu, Weihang Zhang, and Huiqi Li. Absolute-unified multi-class anomaly detection via class-agnostic distribution alignment. *arXiv preprint arXiv:2404.00724*, 2024. 8

[28] Ahmed Hamada. Br35H: Brain tumor detection 2020, 2020. 1

[29] Jiaming Han, Kaixiong Gong, Yiyuan Zhang, Jiaqi Wang, Kaipeng Zhang, Dahua Lin, Yu Qiao, Peng Gao, and Xiangyu Yue. OneLLM: One framework to align all modalities with language. In *Proc. CVPR*, 2024. 8

[30] Haoyang He, Yuhu Bai, Jiangning Zhang, Qingdong He, Hongxu Chen, Zhenye Gan, Chengjie Wang, Xiangtai Li, Guanzhong Tian, and Lei Xie. MambaAD: Exploring state space models for multi-class unsupervised anomaly detection. In *Proc. NeurIPS*, 2024. 2, 8

[31] Haoyang He, Jiangning Zhang, Hongxu Chen, Xuhai Chen, Zhishan Li, Xu Chen, Yabiao Wang, Chengjie Wang, and Lei Xie. A diffusion-based framework for multi-class anomaly detection. In *Proc. AAAI*, 2024. 8

[32] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proc. CVPR*, 2016. 4

[33] Liren He, Zhengkai Jiang, Jinlong Peng, Liang Liu, Qiangang Du, Xiaobin Hu, Wenbing Zhu, Mingmin Chi, Yabiao Wang, and Chengjie Wang. Learning unified reference representation for unsupervised multi-class anomaly detection. In *Proc. ECCV*, 2024. 8

[34] Chih-Hui Ho, Kuan-Chuan Peng, and Nuno Vasconcelos. Long-tailed anomaly detection with learnable class names. In *Proc. CVPR*, 2024. 1, 2, 3, 5, 6, 7, 8, 4, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21

[35] Junjie Hu, Yuanyuan Chen, and Zhang Yi. Automated segmentation of macular edema in oct using deep neural networks. *Medical image analysis*, 2019. 5

[36] Chaoqin Huang, Haoyan Guan, Aofan Jiang, Ya Zhang, Michael Spratling, and Yan-Feng Wang. Registration based few-shot anomaly detection. In *Proc. ECCV*, 2022. 6, 5, 7, 8, 9, 10, 11, 12, 13, 14

[37] 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 *Proc. CVPR*, 2024. 1, 2, 8

[38] Jeeho Hyun, Sangyun Kim, Giyoung Jeon, Seung Hwan Kim, Kyunghoon Bae, and Byung Jun Kang. ReConPatch: Contrastive patch representation learning for industrial anomaly detection. In *Proc. WACV*, 2024. 8

[39] Brian KS Isaac-Medina, Yona Falinie A Gaus, Neelanjan Bhowmik, and Toby P Breckon. Towards open-world object-based anomaly detection via self-supervised outlier synthesis. In *Proc. ECCV*, 2024. 8

[40] Jongheon Jeong, Yang Zou, Taewan Kim, Dongqing Zhang, Avinash Ravichandran, and Onkar Dabeer. WinCLIP: Zero-/few-shot anomaly classification and segmentation. In *Proc. CVPR*, 2023. 1, 2, 5, 8

[41] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *Proc. ICML*, 2021. 8, 2, 4

[42] Pranita Balaji Kanade and PP Gumaste. Brain tumor detection using MRI images. *Brain*, 2015. 1

[43] Felipe Campos Kitamura. Head CT - hemorrhage, 2018. 1

[44] Hyunseo Koh, Dahyun Kim, Jung-Woo Ha, and Jonghyun Choi. Online continual learning on class incremental blurry task configuration with anytime inference. In *Proc. ICLR*, 2022. 8

[45] Bennett Landman, Zhoubing Xu, J Igelsias, Martin Styner, Thomas Langerak, and Arno Klein. Miccai multi-atlas labeling beyond the cranial vault–workshop and challenge. In *Proc. MICCAI Multi-Atlas Labeling Beyond Cranial Vault—Workshop Challenge*, 2015. 5

[46] Joo Chan Lee, Taejune Kim, Eunbyung Park, Simon S Woo, and Jong Hwan Ko. Continuous memory representation for anomaly detection. In *Proc. ECCV*, 2024. 8

[47] Jan Lehr, Jan Philipps, Alik Sargsyan, Martin Pape, and Jörg Krüger. AD3: Introducing a score for anomaly detection dataset difficulty assessment using VIADUCT dataset. In *Proc. ECCV*, 2024. 1

[48] Chun-Liang Li, Kihyuk Sohn, Jinsung Yoon, and Tomas Pfister. CutPaste: Self-supervised learning for anomaly detection and localization. In *Proc. CVPR*, 2021. 8

[49] Shengze Li, Jianjian Cao, Peng Ye, Yuhan Ding, Chongjun Tu, and Tao Chen. ClipSAM: CLIP and SAM collaboration for zero-shot anomaly segmentation. *arXiv preprint arXiv:2401.12665*, 2024. 2, 8

[50] Xurui Li, Ziming Huang, Feng Xue, and Yu Zhou. MuSc: Zero-shot industrial anomaly classification and segmentation with mutual scoring of the unlabeled images. In *Proc. ICLR*, 2024. 8

[51] Xiaofan Li, Zhizhong Zhang, Xin Tan, Chengwei Chen, Yanyun Qu, Yuan Xie, and Lizhuang Ma. PromptAD: Learning prompts with only normal samples for few-shot anomaly detection. In *Proc. CVPR*, 2024. 1, 2, 5, 8

[52] Yuanwei Li, Elizaveta Ivanova, and Martins Bruveris. FADE: Few-shot/zero-shot anomaly detection engine using large vision-language model. In *Proc. BMVC*, 2024. 1, 2, 8

[53] Jingyi Liao, Xun Xu, Manh Cuong Nguyen, Adam Goodge, and Chuan Sheng Foo. COFT-AD: Contrastive fine-tuning for few-shot anomaly detection. *IEEE TIP*, 2024. 8

[54] Chieh Liu, Yu-Min Chu, Ting-I Hsieh, Hwann-Tzong Chen, and Tyng-Luh Liu. Learning diffusion models for multi-view anomaly detection. In *Proc. ECCV*, 2024. 8

[55] Jiaqi Liu, Kai Wu, Qiang Nie, Ying Chen, Bin-Bin Gao, Yong Liu, Jinbao Wang, Chengjie Wang, and Feng Zheng. Unsupervised continual anomaly detection with contrastively-learned prompt. In *Proc. AAAI*, 2024. 1, 8

[56] Ruiying Lu, Bo Chen, Jianqiao Sun, Wenchao Chen, Penghui Wang, Yuanwei Chen, Hongwei Liu, and Pramod K Varshney. Heterogeneity-aware recurrent neural network for hyperspectral and multispectral image fusion. *IEEE Journal of Selected Topics in Signal Processing*, 2022. 8

[57] 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. In *Proc. NeurIPS*, 2023. 1, 2, 4, 5, 6, 8, 7, 9, 10, 11, 12, 13, 14

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

[59] Sergio Naval Marimont and Giacomo Tarroni. Anomaly detection through latent space restoration using vector quantized variational autoencoders. In *Proc. ISBI*, 2021. 1, 5, 8

[60] Saeed Masoudnia and Reza Ebrahimpour. Mixture of experts: a literature survey. *Artificial Intelligence Review*, 2014. 8

[61] Declan GD McIntosh and Alexandra Branzan Albu. Unsupervised, online and on-the-fly anomaly detection for non-stationary image distributions. In *Proc. ECCV*, 2024. 1, 2, 8

[62] Shiyuan Meng, Wenchao Meng, Qihang Zhou, Shizhong Li, Weiye Hou, and Shibo He. MoEAD: A parameter-efficient model for multi-class anomaly detection. In *Proc. ECCV*, 2024. 2, 6, 8, 5, 7, 9, 10, 11, 12, 13, 14

[63] 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 *Proc. ICLR*, 2023. 8

[64] OpenAI. ChatGPT, 2023. 4, 8, 2

[65] Guansong Pang, Chunhua Shen, Longbing Cao, and Anton Van Den Hengel. Deep learning for anomaly detection: A review. *ACM computing surveys (CSUR)*, 2021. 1

[66] Sukanya Patra and Souhaib Ben Taieb. Revisiting deep feature reconstruction for logical and structural industrial anomaly detection. *TMLR*, 2024. 8

[67] Zhen Qu, Xian Tao, Mukesh Prasad, Fei Shen, Zhengtao Zhang, Xinyi Gong, and Guiguang Ding. VCP-CLIP: A visual context prompting model for zero-shot anomaly segmentation. In *Proc. ECCV*, 2024. 1, 2, 8

[68] 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 *Proc. ICML*, 2021. 8

[69] Siddeshwar Raghavan, Jiangpeng He, and Fengqing Zhu. DELTA: Decoupling long-tailed online continual learning. In *Proc. CVPR*, 2024. 8

[70] Danilo Rezende and Shakir Mohamed. Variational inference with normalizing flows. In *Proc. ICML*, 2015. 8

[71] Carlos Riquelme, Joan Puigcerver, Basil Mustafa, Maxim Neumann, Rodolphe Jenatton, André Susano Pinto, Daniel Keysers, and Neil Houlsby. Scaling vision with sparse mixture of experts. In *Proc. NeurIPS*, 2021. 8

[72] Karsten Roth, Latha Pemula, Joaquin Zepeda, Bernhard Schölkopf, Thomas Brox, and Peter Gehler. Towards total recall in industrial anomaly detection. In *Proc. CVPR*, 2022. 8

[73] Marco Rudolph, Bastian Wandt, and Bodo Rosenhahn. Same same but differnet: Semi-supervised defect detection with normalizing flows. In *Proc. WACV*, 2021. 8

[74] Radhwan AA Saleh, Mehmet Zeki Konyar, Kaplan Kaplan, and H Metin Ertunç. Tire defect detection model using machine learning. In *Proc. eSmarTA*, 2022. 1

[75] Mohammadreza Salehi, Niousha Sadjadi, Soroosh Baselizadeh, Mohammad H Rohban, and Hamid R Rabiee. Multiresolution knowledge distillation for anomaly detection. In *Proc. CVPR*, 2021. 8, 6, 7

[76] Jian Shi, Pengyi Zhang, Ni Zhang, Hakim Ghazzai, and Peter Wonka. Dissolving is amplifying: Towards fine-grained anomaly detection. In *Proc. ECCV*, 2024. 8

[77] Luc PJ Sträter, Mohammadreza Salehi, Efstratios Gavves, Cees GM Snoek, and Yuki M Asano. GeneralAD: Anomaly detection across domains by attending to distorted features. In *Proc. ECCV*, 2024. 8

[78] Mingxing Tan and Quoc Le. EfficientNet: Rethinking model scaling for convolutional neural networks. In *Proc. ICML*, 2019. 2

[79] Jiaqi Tang, Hao Lu, Xiaogang Xu, Ruizheng Wu, Sixing Hu, Tong Zhang, Tsz Wa Cheng, Ming Ge, Ying-Cong Chen, and Fugee Tsung. An incremental unified framework for small defect inspection. In *Proc. ECCV*, 2024. 1, 8

[80] Tran Dinh Tien, Anh Tuan Nguyen, Nguyen Hoang Tran, Ta Duc Huy, Soan Duong, Chanh D Tr Nguyen, and Steven QH Truong. Revisiting reverse distillation for anomaly detection. In *Proc. CVPR*, 2023. 8

[81] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. LLaMA: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023. 8

[82] Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. In *Proc. NeurIPS*, 2017. 5, 8

[83] Liyuan Wang, Xingxing Zhang, Hang Su, and Jun Zhu. A comprehensive survey of continual learning: theory, method and application. *IEEE TPAMI*, 2024. 8

[84] Shenxing Wei, Xing Wei, Zhiheng Ma, Shaochen Zhang Songlin Dong, and Yihong Gong. Few-shot online anomaly detection and segmentation. *Knowledge-Based Systems*, 2024. 1, 2, 8

[85] Matthias Wieler and Tobias Hahn. Weakly supervised learning for industrial optical inspection. In *DAGM symposium in*, 2007. 5, 3, 4

[86] Shuai Yang, Zhifei Chen, Pengguang Chen, Xi Fang, Yixun Liang, Shu Liu, and Yingcong Chen. Defect spectrum: A granular look of large-scale defect datasets with rich semantics. In *Proc. ECCV*, 2024. 1

[87] Hang Yao, Ming Liu, Haolin Wang, Zhicun Yin, Zifei Yan, Xiaopeng Hong, and Wangmeng Zuo. GLAD: Towards better reconstruction with global and local adaptive diffusion models for unsupervised anomaly detection. In *Proc. ECCV*, 2024. 8

[88] Xincheng Yao, Ruoqi Li, Zefeng Qian, Lu Wang, and Chongyang Zhang. Hierarchical gaussian mixture normalizing flow modeling for unified anomaly detection. In *Proc. ECCV*, 2024.

[89] Zhiyuan You, Lei Cui, Yujun Shen, Kai Yang, Xin Lu, Yu Zheng, and Xinyi Le. A unified model for multi-class anomaly detection. In *Proc. NeurIPS*, 2022. 5, 6, 8, 7, 9, 10, 11, 12, 13, 14

[90] 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. 8

[91] Vitjan Zavrtanik, Matej Kristan, and Danijel Skočaj. DRAEM-a discriminatively trained reconstruction embedding for surface anomaly detection. In *Proc. ICCV*, 2021. 8, 6, 7

[92] Chongsheng Zhang, George Almpanidis, Gaojuan Fan, Binquan Deng, Yanbo Zhang, Ji Liu, Aouaidjia Kamel, Paolo Soda, and João Gama. A systematic review on long-tailed learning. *arXiv preprint arXiv:2408.00483*, 2024. 1

[93] Gongjie Zhang, Kaiwen Cui, Tzu-Yi Hung, and Shijian Lu. Defect-GAN: High-fidelity defect synthesis for automated defect inspection. In *Proc. WACV*, 2021. 8

[94] 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 preprint arXiv:2311.02612*, 2023. 2, 8

[95] Jiangning Zhang, Xuhai Chen, Yabiao Wang, Chengjie Wang, Yong Liu, Xiangtai Li, Ming-Hsuan Yang, and Dacheng Tao. Exploring plain ViT reconstruction for multi-class unsupervised anomaly detection. In *Proc. IJCAI Workshop on Anomaly Detection with Foundation Models*, 2024. 5, 8, 1

[96] Jiangning Zhang, Haoyang He, Zhenye Gan, Qingdong He, Yuxuan Cai, Zhucun Xue, Yabiao Wang, Chengjie Wang, Lei Xie, and Yong Liu. ADer: A comprehensive benchmark for multi-class visual anomaly detection. *arXiv preprint arXiv:2406.03262*, 2024. 1, 4

[97] Xuan Zhang, Shiyu Li, Xi Li, Ping Huang, Jiulong Shan, and Ting Chen. DeSTSeg: Segmentation guided denoising student-teacher for anomaly detection. In *Proc. CVPR*, 2023. 8

[98] Yiyuan Zhang, Kaixiong Gong, Kaipeng Zhang, Hongsheng Li, Yu Qiao, Wanli Ouyang, and Xiangyu Yue. Meta-Transformer: A unified framework for multimodal learning. *arXiv preprint arXiv:2307.10802*, 2023. 8

[99] Ying Zhao. OmniAL: A unified cnn framework for unsupervised anomaly localization. In *Proc. CVPR*, 2023. 8

[100] Qihang Zhou, Guansong Pang, Yu Tian, Shibo He, and Jiming Chen. AnomalyCLIP: Object-agnostic prompt learning for zero-shot anomaly detection. In *Proc. ICLR*, 2024. 1, 2, 8

[101] Jiawen Zhu and Guansong Pang. Toward generalist anomaly detection via in-context residual learning with few-shot sample prompts. In *Proc. CVPR*, 2024. 8

[102] Jiawen Zhu, Yew-Soon Ong, Chunhua Shen, and Guansong Pang. Fine-grained abnormality prompt learning for zero-shot anomaly detection. *arXiv preprint arXiv:2410.10289*, 2024. 1, 8

[103] Yang Zou, Jongheon Jeong, Latha Pemula, Dongqing Zhang, and Onkar Dabeer. Spot-the-difference self-supervised pre-training for anomaly detection and segmentation. In *Proc. ECCV*, 2022. 1, 5, 3, 4

Appendix

The appendix is organized as follows:

- In Sec. A1, we provide additional implementation and training details of our method.
- In Sec. A2, we provide more details of our new benchmark: long-tailed online AD.
- In Sec. A3, we provide additional results on UniMedical [4, 95] and LTAD [34] benchmarks.
- In Sec. A4, we provide additional results on LTOAD benchmarks.

A1. Details of LTOAD

A1.1. Training details

Reconstruction module R. We detail the training pipeline and details of R introduced in Sec. 4.2. The pipeline is shown in Fig. A1.

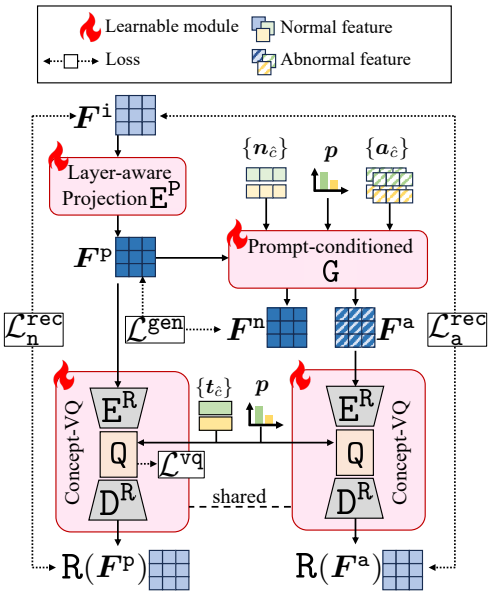


Figure A1. **Reconstruction module** R. Our Concept-VQ autoencoder consists of an encoder $\mathtt{E}^{\mathtt{R}}$, a decoder $\mathtt{D}^{\mathtt{R}}$, and quantization modules Q. It learns to reconstruct input features. As data augmentation, we use G to generate pseudo normal $\boldsymbol{F}^{\mathtt{n}}$ and pseudo abnormal $\boldsymbol{F}^{\mathtt{a}}$ conditioned on prompts $\{\boldsymbol{n}_{\hat{c}}\}$ and $\{\boldsymbol{a}_{\hat{c}}\}$, respectively.

We maximize the similarly of $\boldsymbol{F}^{\mathtt{i}}$ and reconstructed features $\boldsymbol{F}^{\mathtt{r}} = \mathtt{R}(\boldsymbol{F}^{\mathtt{p}})$. At the same time, the reconstruction of $\boldsymbol{F}^{\mathtt{a}}$ should also be as close to $\boldsymbol{F}^{\mathtt{i}}$ as possible, *i.e.*, our reconstruction loss $\mathcal{L}^{\mathtt{rec}}$ is defined as:

$$\mathcal{L}^{\mathtt{rec}} = \underbrace{\mathtt{GAP}\left(\mathbf{1} - \left\langle \boldsymbol{F}^{\mathtt{i}},\ \mathtt{R}(\boldsymbol{F}^{\mathtt{p}})\right\rangle\right)}_{\mathcal{L}^{\mathtt{rec}}_{\mathtt{n}}} + \underbrace{\mathtt{GAP}\left(\mathbf{1} - \left\langle \boldsymbol{F}^{\mathtt{i}},\ \mathtt{R}(\boldsymbol{F}^{\mathtt{a}})\right\rangle\right)}_{\mathcal{L}^{\mathtt{rec}}_{\mathtt{a}}}. \tag{A5}$$

In addition to $\mathcal{L}^{\mathtt{rec}}_{\mathtt{a}}$ in Eq. (A5) which trains G, we also push the similarities between pseudo-normal features and the input normal features using a generator loss $\mathcal{L}^{\mathtt{gen}} = \mathtt{GAP}\left(\mathbf{1} - \langle \boldsymbol{F}^{\mathtt{p}},\ \boldsymbol{F}^{\mathtt{n}}\rangle\right)$.

Semantics module S. We detail the training pipeline and details of S introduced in Sec. 4.3. The pipeline is shown in Fig. A2.

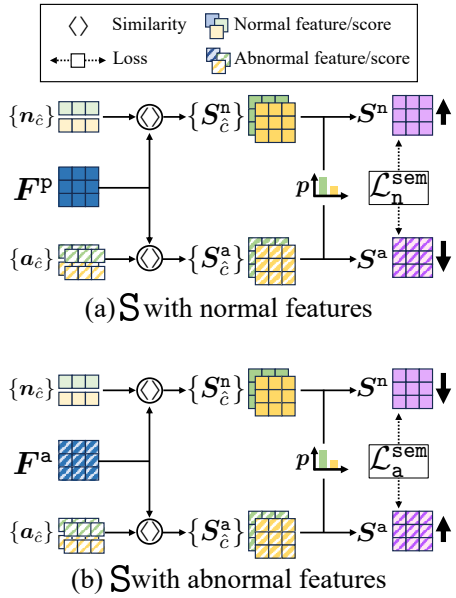


Figure A2. **Semantics module** S. (a) Given normal features $\boldsymbol{F}^{\mathtt{p}}$, $\mathcal{L}^{\mathtt{sem}}_{\mathtt{n}}$ maximizes $\boldsymbol{S}^{\mathtt{n}} - \boldsymbol{S}^{\mathtt{a}}$. Here $\boldsymbol{S}^{\mathtt{n}}$ is the overall per-pixel similarity between $\boldsymbol{F}^{\mathtt{p}}$ and $\{\boldsymbol{n}_{\hat{c}}\}$ while $\boldsymbol{S}^{\mathtt{a}}$ is the one between $\boldsymbol{F}^{\mathtt{p}}$ and $\{\boldsymbol{a}_{\hat{c}}\}$. (b) On the other hand, given abnormal features $\boldsymbol{F}^{\mathtt{a}}$, $\mathcal{L}^{\mathtt{sem}}_{\mathtt{a}}$ maximizes $\boldsymbol{S}^{\mathtt{a}} - \boldsymbol{S}^{\mathtt{n}}$.

We maximize the difference between normal semantic scores $\boldsymbol{S}^{\mathtt{n}}$ and abnormal semantic scores $\boldsymbol{S}^{\mathtt{a}}$ for normal $\boldsymbol{F}^{\mathtt{p}}$ and vice versa for pseudo-abnormal $\boldsymbol{F}^{\mathtt{a}}$. We modify the explicit anomaly margin [51] loss for S. Specifically, the loss is defined as $\mathcal{L}^{\mathtt{sem}}_{\mathtt{n}} = \max(-\delta,\ \mathtt{S}(\boldsymbol{F}^{\mathtt{p}}))$, where $\max(\cdot)$ is a clipping function and $\delta \in \mathbb{R}^{+}$ is a margin term we introduce to further encourage the gap. In contrast, pseudo-abnormal $\boldsymbol{F}^{\mathtt{a}}$ is trained by minimizing $\mathcal{L}^{\mathtt{sem}}_{\mathtt{a}} = \max(-\delta,\ -\mathtt{S}(\boldsymbol{F}^{\mathtt{a}}))$ as shown in Fig. A2(b). Together, our semantics loss $\mathcal{L}^{\mathtt{sem}}$ is defined as the sum of the aforesaid two terms, *i.e.*, $\mathcal{L}^{\mathtt{sem}} = \mathcal{L}^{\mathtt{sem}}_{\mathtt{n}} + \mathcal{L}^{\mathtt{sem}}_{\mathtt{a}}$. The final loss function $\mathcal{L}$ for our offline learning is

$$\mathcal{L} = \lambda_{\mathtt{vq}}\mathcal{L}^{\mathtt{vq}} + \lambda_{\mathtt{rec}}\mathcal{L}^{\mathtt{rec}} + \lambda_{\mathtt{gen}}\mathcal{L}^{\mathtt{gen}} + \lambda_{\mathtt{sem}}\mathcal{L}^{\mathtt{sem}}, \tag{A6}$$

where the λs control the weight of each individual loss.

A1.2. Concept codebooks

As mentioned in Sec. 4.2, we initialized $\boldsymbol{B}_{l,\hat{c}}$ by sampling M codes around its correspondent $\boldsymbol{t}_{\hat{c}}$. Specifically, we initialize $\boldsymbol{B}_{l,\hat{c}} = \boldsymbol{N}_{l,\hat{c}} + \boldsymbol{t}_{\hat{c}}$ where $\boldsymbol{N}_{l,\hat{c}} \in \mathbb{R}^{M \times D}$ is sampled from a Gaussian distribution $\mathcal{N}(\mathbf{0},\ \sigma)$. Throughout

the training, the center of $\boldsymbol{B}_{l,\widehat{c}}$ is always set to $\boldsymbol{t}_{\widehat{c}}$ and we learn the deviation $\boldsymbol{N}_{l,\widehat{c}}$ to fit the data representation. We set $\alpha = 0.02$.

For each pixel location $(h,\ w)$, we find $\boldsymbol{Q}_{l,\widehat{c}}[h,\ w] = \boldsymbol{B}_{l,\widehat{c}}[m^*]$ where $m^* = \arg_m \min \langle \boldsymbol{Z}_l[h,\ w],\ \boldsymbol{B}_{l,\widehat{c}}[m] \rangle$. We aggregate $\mathcal{L}^{\texttt{vq}}_{l,\widehat{c}}$ to have $\mathcal{L}^{\texttt{vq}} = \sum_{l,\widehat{c}} p_{\widehat{c}} \mathcal{L}^{\texttt{vq}}_{l,\widehat{c}}$, where $p_{\widehat{c}}$ is the element of $\boldsymbol{p}$ associated with $\widehat{c}$. The final outputs of our quantization modules are the weighted sum from each concept, *i.e.*, $\sum_{\widehat{c}} p_{\widehat{c}} \boldsymbol{Q}_{l,\widehat{c}}$ for each $l \in [1,\ L]$.

A1.3. Prompt-conditioned data augmentation

We detail the architecture of $\texttt{G}$ introduced in Sec. 4.2. Given the input visual feature map $\boldsymbol{F}^{\texttt{p}} \in \mathbb{R}^{h \times w \times d^{\texttt{f}}}$, where $d^{\texttt{f}}$ is the output dimension of text encoder $\texttt{E}^{\texttt{T}}$, and a set of prompt features, $\texttt{G}$ generate pseudo-normal $\boldsymbol{F}^{\texttt{n}}$ or pseudo-abnormal feature map $\boldsymbol{F}^{\texttt{a}} \in \mathbb{R}^{h \times w \times d^{\texttt{f}}}$ depending on whether the prompt features are $\boldsymbol{n}$ or $\boldsymbol{a}$.

The normal prompt features $\boldsymbol{n} \in \mathbb{R}^{K \times d^{\texttt{f}}}$ and abnormal prompt features $\boldsymbol{a} \in \mathbb{R}^{K \times 5 \times d^{\texttt{f}}}$ contains multiple features from all concepts. We randomly select one feature as the input feature for text-condition, denoted as $\boldsymbol{g}$.

We first pixel-wise concatenate $\boldsymbol{F}^{\texttt{p}}$ and $\boldsymbol{g}$. We feed it to the 5-layer CNN with a hidden dimension of $d^{\texttt{h}}$. We use ReLU as activation layers. We also insert batch normalization layers in between.

A1.4. Semantics similarity scores

We detail how we acquire the overall abnormal similarity map $\boldsymbol{S}^{\texttt{n}}$ in Sec. 4.3. During training, we pick the "hardest" abnormal prompts to enhance the gap between normal and abnormal semantics. Specifically, for $\boldsymbol{F}^{\texttt{p}}$, the hardest abnormal prompt is the one with the highest similarity score since it will most likely confuse the model. In other words, we pick $\boldsymbol{S}^{\texttt{a}}_{\widehat{c}} = \texttt{maximum}(\{\langle \boldsymbol{F}^{\texttt{p}},\ \boldsymbol{a}_{\widehat{c},\,i} \rangle\}_{i=1,\,\cdots,\,5})$. Here $\texttt{maximum}(\cdot)$ picks the largest value on the i-dimension. On the other hand, for $\boldsymbol{F}^{\texttt{a}}$, the hardest abnormal samples are the ones with the lowest similarity score, *i.e.*, $\boldsymbol{S}^{\texttt{a}}_{\widehat{c}} = \texttt{minimum}(\{\langle \boldsymbol{F}^{\texttt{a}},\ \boldsymbol{a}_{\widehat{c},i} \rangle\}_{i=1,\,\cdots,\,5})$. Here $\texttt{minimum}(\cdot)$ picks the smallest value on the i-dimension. Finally, the overall abnormal similarity map is $\boldsymbol{S}^{\texttt{a}} = \sum_{\widehat{c}} p_{\widehat{c}} \boldsymbol{S}^{\texttt{a}}_{\widehat{c}}$.

During inference, we pick the most probable $\boldsymbol{S}^{\texttt{a}}_{\widehat{c}}$, *i.e.* $\boldsymbol{S}^{\texttt{a}}_{\widehat{c}} = \texttt{maximum}(\{\langle \boldsymbol{F}^{\texttt{p}},\ \boldsymbol{a}_{\widehat{c},\,i} \rangle\}_{i=1,\,\cdots,\,5})$.

A1.5. Implementation details

For our foundation model $\texttt{E}^{\texttt{I}}$ and $\texttt{E}^{\texttt{T}}$, we use the ALIGN [41] implementation from Hugging Face Transformer [17], which is open source and publicly available (Apache 2.0). Their $\texttt{E}^{\texttt{I}}$ is based on `efficientnet_b7` [78] containing multiple EfficientNet blocks. We use the same setting in [34] and define our $\boldsymbol{F}^{\texttt{i}}$ as the concatenation of output feature maps from the following blocks: the $3^{\texttt{rd}}$, $10^{\texttt{th}}$, $17^{\texttt{th}}$, and $37^{\texttt{th}}$.

We sample $M = 16$ codes of dimension 640 for each codebook $\boldsymbol{B}_{l,\widehat{c}}$ while setting the number of codebooks $\widehat{K} = 10$. As a comparison, HVQ [57] samples 512 codes of dimension 256 for their K codebooks. Overall, we require far fewer codes than since $16 \times 10 \times 640 \leq 512K \times 256$ for all $K \geq 1$.

For $\widehat{\boldsymbol{Y}}$ (see Eq. 4), we set $\alpha = 0.3$. In $\mathcal{A}^{\texttt{AA}}$ (see Alg. 1), we set $\gamma = 0.3$, $\beta = 5$, $\tau = 0.2$, and $\mathcal{T}(\widehat{\boldsymbol{Y}}) = 0.95 \cdot r(\widehat{\boldsymbol{Y}})$. For online learning on the same dataset, we clip the gradient norm by $1\text{e}{-3}$. For offline learning and online learning across different datasets, we clip the gradient norm by $1\text{e}{-1}$.

For all experiments, we use AdamW optimizer with a learning rate of $1\text{e}{-4}$. During training, we use the balance sampler for sampling long-tailed data distribution in LTAD [34]. All experiments are conducted on an NVIDIA RTX 6000 GPU.

A1.6. Concepts and prompts initialization

We now provide the initialization of the $\widehat{\mathcal{C}}$ and $\mathcal{P}^{\texttt{a}}$ we used for all datasets.

MVTec. Our $\widehat{\mathcal{C}}$ contains the following vocabularies: *semiconductor*, *zipper*, *beech*, *walnut*, *circuit*, *microscopy*, *mahogany*, *hardwood*, *medicines*, and *antibiotics*. For each $\widehat{c} \in \widehat{\mathcal{C}}$, we acquire the 5 abnormal prompts in $\mathcal{P}^{\texttt{a}}_{\widehat{c}}$ by asking Copilot [13] the following query: "*Out of 100* $[\widehat{c}]$ *which looks generally the same, one appears to have a broken region. List 5 examples.*"

VisA. Our $\widehat{\mathcal{C}}$ contains the following vocabularies:*circuit*, *electronics*, *candles*, *bananas*, *supplements*, *semiconductor*, *sensors*, *vitamin*, *usb*, and *wheel*. For each $\widehat{c} \in \widehat{\mathcal{C}}$, we acquire the 5 abnormal prompts in $\mathcal{P}^{\texttt{a}}_{\widehat{c}}$ by asking Copilot [13] the following query: "*Out of 100* $[\widehat{c}]$ *which looks generally the same, one appears to have a broken region. List 5 examples.*"

DAGM. Our $\widehat{\mathcal{C}}$ contains the following vocabularies: *ultrasound*, *fetal*, *wavelengths*, *topography*, *microscopy*, *imaging*, *irregularities*, *electromagnetic*, *noise*, and *seismic*. For each $\widehat{c} \in \widehat{\mathcal{C}}$, we acquire the 5 abnormal prompts in $\mathcal{P}^{\texttt{a}}_{\widehat{c}}$ by asking Copilot [13] the following query: "*Out of 100* $[\widehat{c}]$ *which looks generally the same, one appears to have a broken region. List 5 examples.*"

Uni-Medical. Our $\widehat{\mathcal{C}}$ contains the following vocabularies: *neuroscience*, *brain*, *vascular*, *microscopy*, *imaging*, *mri*, *mitochondrial*, *neurons*, *ultrasound*, and *cerebral*. For each $\widehat{c} \in \widehat{\mathcal{C}}$, we acquire the 5 abnormal prompts in $\mathcal{P}^{\texttt{a}}_{\widehat{c}}$ by asking ChatGPT [64] the following query: "*Out of 100* $[\widehat{c}]$ *images which look generally the same, one appears to have a broken region. List 5 examples.*"

A2. Long-tailed online AD benchmark

We re-address the design motivation of our long-tailed online AD (LTOAD) benchmark. Then we document the additional details for setting up the LTOAD for the following

datasets, *i.e.*, MVTec [5], VisA [103], and DAGM [85].

Our benchmark focuses on evaluating the performance of $\mathbf{F}_{\theta_t}$ for each step t in online training stream $\mathcal{D}^0$ where θ_0 is pre-trained on an long-tailed $\mathcal{D}^{\mathsf{T}}$, *i.e.*, LTAD [34]. We design $\mathcal{D}^0$ not to be long-tailed because, in the industrial environment, we need quick adaptation in the online learning process. If $\mathcal{D}^0$ is also long-tailed, it is difficult for the model to learn anomaly detection effectively due to insufficient anomaly samples. We will provide all of the used meta files of $\mathcal{D}^{\mathsf{T}}$, $\mathcal{D}^0$, and $\mathcal{D}^{\mathsf{E}}$ along with our code once accepted.

Dataset	$\lvert\mathcal{C}\rvert$	Elements
MVTec	15	hazelnut, leather, bottle, wood, carpet, tile, metal nut, toothbrush, zipper, transistor, grid, pill, capsule, cable, screw
VisA	12	pcb3, pcb2, pcb1, pcb4, macaroni1, macaroni2, candle, cashew, fryum, capsules, chewinggum, pipe fryum, screw
DAGM	10	Class10, Class7, Class9, Class8, Class2, Class3, Class5, Class1, Class4, Class6

Table A1. $\mathcal{C}^{\texttt{head}}$ **and** $\mathcal{C}^{\texttt{tail}}$ **of MVTec, VisA, and DAGM.**

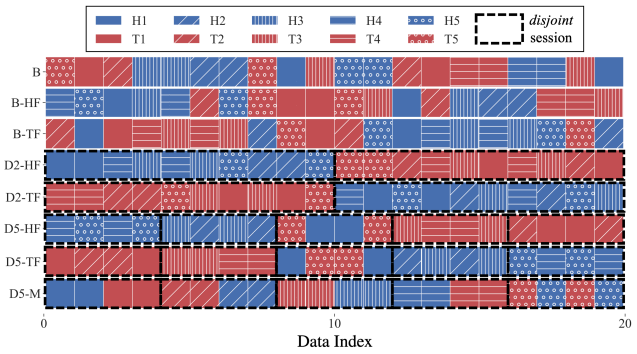


Figure A3. **The configurations for** $\mathcal{D}^0$**.** We define the 8 configurations with combinations of different session type $\in \{blurry, disjoint\}$ and ordering type $\in \{head\text{-}first, head\text{-}first, else\}$. We visualize them (see Tab. 1) with a toy example on $\mathcal{D}^0$. $\mathcal{C}^{\texttt{head}}$ and $\mathcal{C}^{\texttt{tail}}$ each has 5 classes.

In Fig. A3, we visualize the proposed 8 configurations in Tab. 1 with a toy example for better understanding.

A2.1. MVTec

MVTec is an industrial anomaly detection dataset, containing 15 classes. In the $1^{\texttt{st}}$ row of Tab. A1, we list out all the classes and whether each of them belongs to $\mathcal{C}^{\texttt{head}}$ or $\mathcal{C}^{\texttt{tail}}$ in LTAD [34] benchmark. We first detail the selected classes of each *D5* configuration in LTOAD. We then discuss the random function we used to sample *B-HF* and *B-TF* configurations.

In *D5-HF*, we divide the $\mathcal{D}^0$ into 5 disjoint sessions and we put the $\mathcal{C}^{\texttt{head}}$ in the first few sessions. We organize the classes in each session as the following list so that the number of images in each session of $\mathcal{D}^0$ is roughly the same.

1. carpet, hazelnut, metal nut
2. tile, leather, wood
3. bottle, screw, capsule
4. zipper, cable, toothbrush
5. pill, transistor, grid

On the other hand, in *D5-TF*, we divide the $\mathcal{D}^0$ into 5 disjoint sessions and we put the $\mathcal{C}^{\texttt{tail}}$ in the first few sessions. We organize the classes in each session as the following list, *i.e.D5-TF* in reverse order.

1. pill, transistor, grid
2. zipper, cable, toothbrush
3. screw, capsule, bottle
4. tile, leather, wood
5. carpet, hazelnut, metal nut

In *D5-M*, we divide the $\mathcal{D}^0$ into 5 disjoint sessions and we put both $\mathcal{C}^{\texttt{head}}$ and $\mathcal{C}^{\texttt{tail}}$ in each session. We organize the classes in each session as the following list

1. carpet, cable, grid
2. hazelnut, transistor, capsule,
3. metal nut, wood, screw
4. leather, pill, toothbrush
5. bottle, tile, zipper

In *B-HF*, for each $i \in 1, \cdots, N^0$ where $N^0 = |\mathcal{D}^0|$, we use the following operation to determine where $\widetilde{\boldsymbol{X}}_i$ should be sampled from $\mathcal{C}^{\texttt{head}}$ or $\mathcal{C}^{\texttt{tail}}$. Let the number of remaining $\mathcal{C}^{\texttt{head}}$ images at step t, *i.e.* $\mathcal{D}^0[t:]$ be $N_t^{\texttt{head}}$ and the one for $\mathcal{C}^{\texttt{tail}}$ be $N_t^{\texttt{tail}}$. We sample from $\mathcal{U}(0,1)$ repeatedly for r times and keep the largest number as u. If $u \geq \frac{N_t^{\texttt{head}}}{N_t^{\texttt{head}}+N_t^{\texttt{tail}}}$, then we sample $\widetilde{\boldsymbol{X}}_i$ from $\mathcal{C}^{\texttt{head}}$ and otherwise. We set $r = 5$.

In *B-TF*, we use a similar operation. However, we sample from $\mathcal{U}(0,1)$ repeatedly for r times and keep the smallest number as u instead. If $u \leq \frac{N_t^{\texttt{head}}}{N_t^{\texttt{head}}+N_t^{\texttt{tail}}}$, then we sample $\widetilde{\boldsymbol{X}}_i$ from $\mathcal{C}^{\texttt{tail}}$ and otherwise.

A2.2. VisA

VisA is an industrial anomaly detection dataset, containing 12 classes. We list out the $\mathcal{C}$ in the $2^{\texttt{nd}}$ row of Tab. A1.

Here we use the same design logic for the configurations in MVTec. In *D5-HF*, we organize the classes in each session as the following list.

1. pcb3, pcb4
2. pcb1, pcb2
3. macaroni1, macaroni2
4. chewinggum, fryum, pipe fryum
5. candle, capsules, cashew

In *D5-TF*, we organize the classes in each session as the following list.

1. candle, capsules, cashew
2. chewinggum, fryum, pipe fryum
3. macaroni1, macaroni2

4. pcb1, pcb2
5. pcb3, pcb4

In *D5-M*, we organize the classes in each session as the following list.

1. candle, macaroni1
2. capsules, macaroni2, pcb1
3. cashew, pcb2
4. chewinggum, pcb3
5. fryum, pipe fryum, pcb4

A2.3. DAGM

DAGM is a synthetic anomaly detection dataset, containing 10 classes. We list out the $\mathcal{C}$ in the 3$^{\text{rd}}$ row of Tab. A1. Each class has a different synthetic pattern.

Here we use the same design logic for the configurations in MVTec. In *D5-HF*, we organize the classes in each session as the following list.

1. Class9, Class10
2. Class7, Class8
3. Class2 Class6,
4. Class4, Class5
5. Class1, Class3

In *D5-TF*, we organize the classes in each session as the following list.

1. Class1, Class3
2. Class4, Class5
3. Class6, Class2
4. Class7, Class8
5. Class9, Class10

In *D5-M*, we organize the classes in each session as the following list.

1. Class2, Class1
2. Class7, Class3
3. Class8, Class4
4. Class9, Class5
5. Class10, Class6

A3. Offline experiments

We provide additional analysis of our approach. We then report the performances on $\mathcal{C}^{\texttt{head}}$, on $\mathcal{C}^{\texttt{tail}}$, and on each $c \in \mathcal{C}$ in the following tables under various datasets and long-tailed imbalance [34] settings.

A3.1. Additional analysis

Alternative approach to our concept learning. A naive baseline for our concept learning is to replace the VLM encoder, *i.e*. ALIGN [41], with simpler vision architectures like ResNet [32] using clustering methods. However, this alternative approach loses all the language information for each class and removes the capability of the semantics module S. Additionally, long-tailed data distribution is challenging for simple clustering methods. We provide the empirical comparison on MVTec *exp100*. While LTOAD achieves image-level AUROC of 85.26 for Det. , the ResNet-based K-means alternative only achieves 57.16.

Computational cost. Our method does not introduce computational overhead. We pre-compute the $\widehat{C}$ before training and it takes only 3 minutes to run on one single NVIDIA RTX 6000 GPU. Additionally, we use fewer parameters than LTAD [34] since our $|\widehat{C}| < |\mathcal{C}|$ thus requires fewer learned prompts. Lastly, the throughput of LTOAD is 61.74 fps while LTAD's is 32.78.

A3.2. MVTec, VisA, and DAGM

Experiment setup. We follow the offline LTAD [34] benchmarks and report on MVTec [5], VisA [103], and DAGM [85].

Additional quantitative comparison. The results are in the corresponding tables:

- MVTec
  1. *exp100*: Tab. A11 and Tab. A12.
  2. *exp200*: Tab. A13 and Tab. A14.
  3. *step100*: Tab. A15 and Tab. A16.
  4. *step200*: Tab. A17 and Tab. A18.
- VisA
  1. *exp100*: Tab. A19 and Tab. A20.
  2. *exp200*: Tab. A21 and Tab. A22.
  3. *exp500*: Tab. A23 and Tab. A24.
  4. *step100*: Tab. A25 and Tab. A26.
  5. *step200*: Tab. A27 and Tab. A28.
  6. *step500*: Tab. A29 and Tab. A30.
- DAGM
  1. Overall: Tab. A3.
  2. *exp50*: Tab. A31 and Tab. A32.
  3. *exp100*: Tab. A33 and Tab. A34.
  4. *exp200*: Tab. A35 and Tab. A36.
  5. *exp500*: Tab. A37 and Tab. A38.
  6. *reverse exp200*: Tab. A39 and Tab. A40.
  7. *step50*: Tab. A41 and Tab. A42.
  8. *step100*: Tab. A43 and Tab. A44.
  9. *step200*: Tab. A45 and Tab. A46..
  10. *step500*: Tab. A47 and Tab. A48..
  11. *reverse step200*: Tab. A49 and Tab. A50..

We note that in *reverse exp200* and *reverse step200*, LTAD switches the $\mathcal{C}^{\texttt{head}}$ and $\mathcal{C}^{\texttt{tail}}$. We observe a consistent improvement of LTOAD in comparison to all baselines on most configurations. Notably, LTOAD excels on the more challenging $\mathcal{C}^{\texttt{tail}}$ which is the focus of long-tailed settings.

Additional Qualitative Comparison. Finally, we show additional qualitative results in Fig. A4.

More evaluation metrics.. We report more evaluation metrics on MVTec following [96], *i.e*., pixel-level AUPRO (AUPRO$_{\text{P}}$), mean average precision (mAP), and mean F1-score (mF1-max), for comprehensive comparisons in Tab. A2. We observe consistent improvement under all metrics.

Method	*exp100*			*exp200*			*step100*			*step200*		
	mAP	mF1-max	AUPRO_P	mAP	mF1-max	AUPRO_P	mAP	mF1-max	AUPRO_P	mAP	mF1-max	AUPRO_P
MoEAD [62]	35.63	41.19	82.72	33.87	39.74	80.77	33.50	39.17	80.54	31.55	36.86	77.88
LTOAD	**38.21**	**43.86**	**85.98**	**37.57**	**42.65**	**85.30**	**38.22**	**43.32**	**85.71**	**35.29**	**40.45**	**82.86**

Table A2. **Comparison (↑) on MVTec** under more evaluation metrics, *i.e.* mAP, mF1-max, and AUPRO_P.

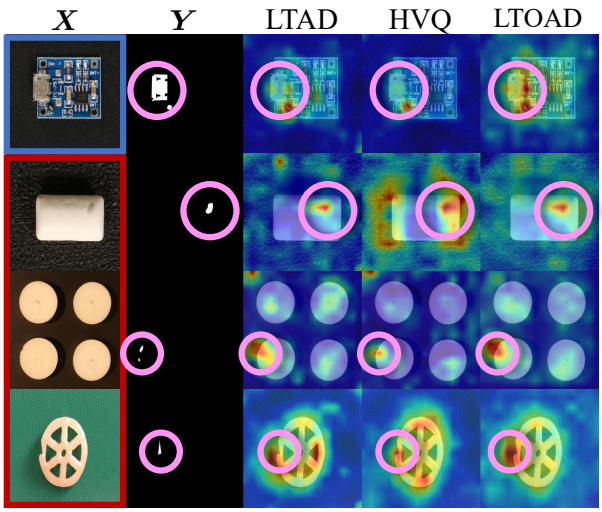


Figure A4. **Qualitative comparison** among LTAD [34], HVQ [57], and LTOAD on VisA offline LTAD [34] *exp100* benchmark. Inputs from $\mathcal{C}^{\texttt{head}}$ / $\mathcal{C}^{\texttt{tail}}$ are outlined in blue / red. Smaller ground truth anomaly masks are circled in pink.

Method	CA	*exp100*		*exp200*		*step100*		*step200*	
		Det.	Seg.	Det.	Seg.	Det.	Seg.	Det.	Seg.
RegAD [36]	✗	84.86	90.29	84.86	90.29	84.86	90.29	84.86	90.29
AnomalyGPT [25]	✗	85.31	77.20	83.29	77.16	86.48	78.76	84.73	78.29
LTAD [34]	✗	94.40	97.30	94.29	**97.19**	93.97	97.07	93.79	96.84
HVQ [57]	✗	75.42	85.54	76.19	85.32	75.78	85.23	75.92	84.63
LTOAD *	✗	94.92	96.81	94.13	96.77	94.18	96.90	93.88	96.64
UniAD [89]	✓	84.34	90.13	83.56	89.73	81.11	89.11	80.33	89.07
MoEAD [62]	✓	73.26	84.99	72.61	84.18	70.83	83.34	70.01	82.27
LTOAD	✓	**95.58**	**97.32**	**94.71**	97.11	**94.83**	**97.30**	**94.12**	**97.14**

Table A3. **Comparison (↑) on offline DAGM** in the same format as Tab. 2

A3.3. Uni-Medical

Experiment setup. We provided additional results on Uni-Medical [4, 95] under long-tailed training settings. Uni-Medical is a medical anomaly detection dataset extracted from BMAD [4]. It consists of the following benchmarks of medical imaging: Brain MRI [3], Liver CT BTCV [6, 45], and Retinal OCT [35]. We use the same design logic of LTAD benchmark [34] and create the long-tailed training configurations for Uni-Medical. Since there are only 3 classes in Uni-Medical, we experiment with cases of each being the head class. We document the configurations of $\mathcal{D}^0$ in Tab. A4.

Config.	$\mathcal{C}^{\texttt{head}}$	Imbalance type	N_c		
			brain	liver	retinal
brain exp100	brain	exponential decay	1542	154	15
brain step100	brain	step function	1542	15	15
liver exp100	liver	exponential decay	15	1542	154
liver step100	liver	step function	15	1542	15
retinal exp100	retinal	exponential decay	154	15	1542
retinal step100	retinal	step function	15	15	1542

Table A4. **Configurtions of $\mathcal{D}^\text{T}$ for long-tailed Uni-Medical.** We denote the number of images per class c as N_c and highlight the N_c of $c \in \mathcal{C}^{\texttt{head}}$ or $\mathcal{C}^{tail}$. We set all imbalance factors to 100.

Baselines & training details.. We compare with the SOTA MoEAD [62] ($\mathcal{G}_1$) and HVQ [57] ($\mathcal{G}_2$). We set $\widehat{K} = 10$. We train LTOAD for 25 epochs on *liver exp100* and *liver step100*. On other configurations, we train for 5 epochs. For detection, we reduce $\widehat{\boldsymbol{Y}}$ to $\widehat{y}$ by calculating its standard deviation.

Quantitative comparison. The results are in Tab. A5-A10 in the corresponding tables:

- *brain exp100*: Tab. A5.
- *brain step100*: Tab. A6.
- *liver exp100*: Tab. A7.
- *liver step100*: Tab. A8.
- *retinal exp100*: Tab. A9.
- *retinal step100*: Tab. A10.

On most configurations, LTOAD outperforms class-agnostic MoEAD while being competitive to class-aware HVQ which requires additional information.

A4. Online experiments

We report additional results on our proposed benchmark LTOAD under more configurations. For each dataset, we report the quantitative comparison at $t = T$ on all $\mathcal{D}^0$ configuration with or without our $\mathcal{A}^{\texttt{AA}}$ in Tab. A51, Tab. A52, and Tab. A53, respectively. We also plot the performance curves of $\mathbf{F}_{\theta_t}$ on $\mathcal{C}^{\texttt{head}}$ and $\mathcal{C}^{\texttt{tail}}$ for each step $t \in [0, \cdots, T]$. We set $T = 33$ for all experiments. Specifically, we list them as the following.

- MVTec
  1. *exp100*: Fig. A5.
  2. *exp200*: Fig. A6.
  3. *step100*: Fig. A7.

Method	CA	$\mathcal{C}$		$\mathcal{C}^{\text{head}}$		$\mathcal{C}^{\text{tail}}$		*brain*		*liver*		*retinal*	
		Det.	Seg.	Det.	Seg.	Det.	Seg.	Det.	Seg.	Det.	Seg.	Det.	Seg.
HVQ [57]	✗	65.21	93.84	83.15	97.03	56.23	92.25	83.15	97.03	45.72	**95.42**	66.74	89.07
MoEAD [62]	✓	60.16	91.09	**86.74**	**97.50**	46.88	87.88	**86.74**	**97.50**	43.60	94.82	50.15	80.95
LTOAD	✓	**72.81**	**94.93**	80.77	96.32	**68.83**	**94.24**	80.77	96.32	**58.05**	95.07	**79.60**	**93.42**

Table A5. **Comparison (↑) on Uni-Medical** *brain exp100* using the same evaluation metrics as Tab. A11 and Tab. A12.

Method	CA	$\mathcal{C}$		$\mathcal{C}^{\text{head}}$		$\mathcal{C}^{\text{tail}}$		*liver*		*retinal*		*brain*	
		Det.	Seg.	Det.	Seg.	Det.	Seg.	Det.	Seg.	Det.	Seg.	Det.	Seg.
HVQ [57]	✗	**72.50**	94.49	**75.10**	94.45	71.20	94.51	**59.10**	**96.92**	83.30	92.11	**75.10**	94.45
MoEAD [62]	✓	59.22	91.44	58.14	92.11	59.76	91.10	52.48	95.40	67.05	86.80	58.14	92.11
LTOAD	✓	69.19	**94.53**	63.91	**94.52**	**71.82**	**94.54**	58.91	95.69	**84.74**	**93.38**	63.91	**94.52**

Table A7. **Comparison (↑) on Uni-Medical** *liver exp100* using the same evaluation metrics as Tab. A5.

Method	CA	$\mathcal{C}$		$\mathcal{C}^{\text{head}}$		$\mathcal{C}^{\text{tail}}$		*retinal*		*brain*		*liver*	
		Det.	Seg.	Det.	Seg.	Det.	Seg.	Det.	Seg.	Det.	Seg.	Det.	Seg.
HVQ [57]	✗	74.80	**95.04**	80.91	**96.14**	71.75	**94.49**	**86.15**	93.19	80.91	**96.14**	57.34	**95.78**
MoEAD [62]	✓	69.87	93.57	78.12	95.27	65.75	92.72	79.55	92.20	78.12	95.27	51.95	93.24
LTOAD	✓	**75.55**	94.49	**83.03**	95.53	**71.81**	93.97	85.88	**94.19**	**83.03**	95.53	**57.75**	93.75

Table A9. **Comparison (↑) on Uni-Medical** *retinal exp100* using the same evaluation metrics as Tab. A5.

Method	CA	$\mathcal{C}$		$\mathcal{C}^{\text{head}}$		$\mathcal{C}^{\text{tail}}$		*brain*		*liver*		*retinal*	
		Det.	Seg.	Det.	Seg.	Det.	Seg.	Det.	Seg.	Det.	Seg.	Det.	Seg.
HVQ [57]	✗	69.23	93.89	**79.60**	**97.22**	64.04	92.23	**79.60**	**97.22**	**57.74**	**95.70**	70.34	88.76
MoEAD [62]	✓	61.11	90.99	78.21	96.88	52.56	88.05	50.88	92.83	54.24	83.26	78.21	**96.88**
LTOAD	✓	**73.44**	**94.61**	78.13	96.08	**71.10**	**93.88**	78.13	96.08	56.31	93.94	**85.89**	93.81

Table A6. **Comparison (↑) on Uni-Medical** *brain step100* using the same evaluation metrics as Tab. A5.

Method	CA	$\mathcal{C}$		$\mathcal{C}^{\text{head}}$		$\mathcal{C}^{\text{tail}}$		*liver*		*retinal*		*brain*	
		Det.	Seg.	Det.	Seg.	Det.	Seg.	Det.	Seg.	Det.	Seg.	Det.	Seg.
HVQ [57]	✗	66.88	93.34	**73.63**	94.31	63.50	92.86	58.94	**96.80**	68.06	88.92	**73.63**	94.31
MoEAD [62]	✓	57.01	90.39	61.12	92.19	54.96	89.49	54.60	95.86	55.32	83.13	61.12	92.19
LTOAD	✓	**70.27**	**94.53**	67.26	**94.42**	**71.77**	**94.58**	**59.19**	95.76	**84.35**	**93.40**	67.26	**94.42**

Table A8. **Comparison (↑) on Uni-Medical** *liver step100* using the same evaluation metrics as Tab. A5.

Method	CA	$\mathcal{C}$		$\mathcal{C}^{\text{head}}$		$\mathcal{C}^{\text{tail}}$		*retinal*		*brain*		*liver*	
		Det.	Seg.	Det.	Seg.	Det.	Seg.	Det.	Seg.	Det.	Seg.	Det.	Seg.
HVQ [57]	✗	**71.44**	**94.41**	**72.07**	93.96	71.13	**94.63**	**87.65**	93.68	**72.07**	93.96	54.61	**95.58**
MoEAD [62]	✓	64.96	92.87	62.23	92.79	66.32	92.91	81.71	92.28	62.23	92.79	50.94	93.55
LTOAD	✓	71.31	93.69	71.11	**94.39**	**71.41**	93.34	86.17	**94.70**	71.11	**94.39**	**56.66**	91.98

Table A10. **Comparison (↑) on Uni-Medical** *retinal step100* using the same evaluation metrics as Tab A5.

Method	CA	$\mathcal{C}$	$\mathcal{C}^{\text{head}}$	$\mathcal{C}^{\text{tail}}$	*hazelnut*	*leather*	*bottle*	*wood*	*carpet*	*tile*	*metal nut*	*toothbrush*	*zipper*	*transistor*	*grid*	*pill*	*capsule*	*cable*	*screw*
MKD [75]	✗	78.92	83.11	75.26	97.00	78.94	98.97	91.05	74.48	78.94	62.37	72.22	90.49	79.33	64.24	78.40	69.01	78.54	69.81
DRAEM [91]	✗	79.57	95.22	65.87	98.32	98.91	99.68	**99.91**	92.17	97.94	79.66	83.61	93.01	62.54	43.02	68.14	41.76	53.31	**81.57**
RegAD [36]	✗	82.43	90.04	70.51	91.84	98.61	99.20	97.86	97.44	97.59	89.77	75.69	85.58	81.68	68.80	69.75	64.67	64.67	53.30
AnomalyGPT [25]	✗	87.44	96.40	79.60	96.21	**100.00**	96.75	88.07	98.19	97.66	97.95	95.00	87.16	79.77	91.98	84.44	67.03	69.33	62.10
LTAD [34]	✗	88.86	99.09	79.90	99.82	**100.00**	99.92	99.30	99.88	99.24	95.50	87.50	89.23	93.79	87.80	84.04	64.82	77.49	54.56
HVQ [57]	✗	87.43	99.34	77.00	99.93	**100.00**	**100.00**	97.54	99.88	98.99	99.02	85.00	**93.67**	89.71	78.61	90.51	70.12	60.04	48.35
UniAD [89]	✓	87.70	99.27	77.58	**100.00**	**100.00**	99.76	99.56	99.79	**99.38**	96.43	88.61	86.65	91.58	92.64	81.58	63.78	63.64	52.22
MoEAD [62]	✓	84.73	98.94	72.29	99.21	**100.00**	99.52	97.54	**100.00**	97.84	98.48	90.28	88.79	89.75	65.16	66.91	66.69	61.15	49.62
LTOAD	✓	**93.42**	**99.45**	**88.14**	**100.00**	**100.00**	99.52	98.51	**100.00**	98.67	**99.46**	**96.39**	91.81	**96.50**	**97.33**	**94.27**	**83.05**	**87.82**	57.98

Table A11. **Comparison (↑) on MVTec** *exp100* [34] in image-level AUROC for anomaly detection (Det.). The column CA (class-agnostic) indicates whether a method requires class names or the number of classes during training or not (require: ✗; not require: ✓). We report the class-wise performance on $\mathcal{C}^{\text{head}}$ and $\mathcal{C}^{\text{tail}}$. The classes are sorted from having the most images (left) to the least (right).

4. *step200*: Fig. A8.

- VisA
  1. *exp100*: Fig. A9.
  2. *exp200*: Fig. A10.
  3. *step100*: Fig. A11.
  4. *step200*: Fig. A12.
- DAGM
  1. *exp100*: Fig. A13.
  2. *exp200*: Fig. A14.
  3. *step100*: Fig. A15.
  4. *step200*: Fig. A16.

Apart from the synthetic DAGM dataset where the offline performance is saturated, we observe that LTOAD improves the performances in most cases, especially under the more challenging offline long-tailed settings, *i.e*. *step200*.

Method	CA	$\mathcal{C}$	$\mathcal{C}^{\text{head}}$	$\mathcal{C}^{\text{tail}}$	*hazelnut*	*leather*	*bottle*	*wood*	*carpet*	*tile*	*metal nut*	*toothbrush*	*zipper*	*transistor*	*grid*	*pill*	*capsule*	*cable*	*screw*
MKD [75]	✗	85.95	87.39	84.69	95.00	96.87	93.10	77.74	93.05	76.84	79.13	93.86	90.14	73.69	72.36	88.75	93.16	71.67	93.95
DRAEM [91]	✗	85.17	93.68	77.73	97.80	97.56	95.18	**97.33**	94.33	**97.36**	76.21	97.34	**97.43**	66.95	76.55	89.00	44.65	59.90	90.08
RegAD [36]	✗	95.20	**96.92**	93.69	98.07	99.15	98.12	95.46	98.67	94.18	94.81	96.88	97.12	92.44	79.05	**97.29**	96.42	**96.42**	**93.96**
AnomalyGPT [25]	✗	89.68	93.64	86.21	92.06	**99.58**	93.42	89.47	**98.97**	94.00	88.02	97.06	93.22	67.18	94.33	75.00	87.84	85.00	89.17
LTAD [34]	✗	94.46	95.96	93.15	98.12	99.29	96.84	90.45	98.52	92.52	95.96	97.91	93.93	96.89	93.76	90.09	95.02	91.67	85.92
HVQ [57]	✗	**95.25**	96.36	94.28	**98.54**	98.80	**98.26**	92.26	98.69	91.55	**96.39**	98.24	96.59	**97.38**	91.72	93.39	**98.17**	90.74	87.98
UniAD [89]	✓	93.95	95.46	92.64	97.83	98.79	95.64	90.66	98.52	91.65	95.18	97.96	92.35	96.38	93.80	88.66	94.68	88.93	88.36
MoEAD [62]	✓	94.34	96.02	92.87	98.00	98.86	98.14	92.21	98.80	90.34	95.78	**98.31**	95.24	96.69	87.33	89.91	97.14	90.59	87.71
LTOAD	✓	95.21	95.64	**94.83**	98.16	98.57	96.69	91.01	98.72	93.57	92.79	98.14	93.24	96.82	**95.14**	93.71	95.86	94.08	91.70

Table A12. **Comparison (↑) on MVTec** *exp100* [34] in pixel-level AUROC for anomaly segmentation (Seg.). The column CA (class-agnostic) indicates whether a method requires class names or the number of classes during training or not (require: ✗; not require: ✓). We report the class-wise performance on $\mathcal{C}^{\text{head}}$ and $\mathcal{C}^{\text{tail}}$. The classes are sorted from having the most images (left) to the least (right).

Method	CA	$\mathcal{C}$	$\mathcal{C}^{\text{head}}$	$\mathcal{C}^{\text{tail}}$	*hazelnut*	*leather*	*bottle*	*wood*	*carpet*	*tile*	*metal nut*	*toothbrush*	*zipper*	*transistor*	*grid*	*pill*	*capsule*	*cable*	*screw*
MKD [75]	✗	79.93	84.47	75.96	96.93	78.19	98.97	91.23	74.36	86.98	64.66	72.78	89.97	79.08	67.84	76.70	70.56	78.84	71.90
DRAEM [91]	✗	78.82	96.39	63.44	97.10	99.89	99.76	**99.91**	95.78	98.05	84.26	80.00	88.07	60.00	46.70	50.19	33.58	54.96	**94.09**
AnomalyGPT [25]	✗	85.80	97.01	76.00	98.57	**100.00**	96.59	89.04	98.92	97.84	98.09	88.61	86.52	77.06	90.14	88.54	62.76	57.01	57.38
LTAD [34]	✗	86.05	98.99	74.73	99.29	99.93	99.76	99.56	99.96	99.39	95.06	86.94	85.56	79.67	88.47	81.91	64.42	58.62	52.22
HVQ [57]	✗	84.90	99.14	72.43	**99.89**	**100.00**	**100.00**	98.16	99.76	**99.93**	96.24	84.44	95.01	66.62	81.45	77.69	68.01	61.08	45.15
UniAD [89]	✓	86.21	99.16	74.89	99.85	**100.00**	**100.00**	97.98	99.75	99.89	96.67	85.83	**96.11**	74.54	87.55	70.07	64.42	57.74	62.88
MoEAD [62]	✓	83.23	98.43	69.93	99.18	**100.00**	99.52	96.93	99.92	96.79	96.68	88.33	88.94	70.13	67.00	72.48	67.09	58.56	46.92
LTOAD	✓	**92.02**	**99.17**	**85.77**	99.79	**100.00**	99.29	97.28	**100.00**	98.67	**99.17**	**94.44**	92.70	**90.29**	**97.66**	**94.16**	**78.58**	**87.28**	51.01

Table A13. **Comparison (↑) on MVTec** *exp200* [34] in image-level AUROC using the same evaluation metrics as Tab. A11.

Method	CA	$\mathcal{C}$	$\mathcal{C}^{\text{head}}$	$\mathcal{C}^{\text{tail}}$	*hazelnut*	*leather*	*bottle*	*wood*	*carpet*	*tile*	*metal nut*	*toothbrush*	*zipper*	*transistor*	*grid*	*pill*	*capsule*	*cable*	*screw*
MKD [75]	✗	86.01	87.15	85.02	94.01	96.81	93.02	77.79	93.25	77.04	78.10	93.53	90.26	75.28	72.88	88.81	93.16	72.45	**93.76**
DRAEM [91]	✗	82.95	93.98	73.30	96.96	99.15	94.34	**95.62**	95.25	**95.34**	81.24	96.43	96.15	59.05	77.78	83.77	38.02	44.28	90.94
AnomalyGPT [25]	✗	90.15	94.01	86.76	92.47	**99.54**	94.34	89.70	**98.96**	94.62	88.48	96.87	92.39	67.64	94.95	76.63	89.01	85.46	91.16
LTAD [34]	✗	94.18	96.25	92.37	97.87	99.04	96.79	92.16	98.65	94.23	95.00	97.83	92.86	93.45	93.85	93.11	95.37	85.92	86.55
HVQ [57]	✗	94.79	**96.32**	93.44	**98.61**	98.78	**98.23**	92.14	98.66	91.53	**96.32**	98.25	**96.16**	93.08	91.34	92.10	**97.86**	91.26	87.47
UniAD [89]	✓	93.26	95.91	90.95	98.03	98.67	97.91	93.67	98.48	91.97	92.68	98.23	96.11	89.44	89.96	86.62	96.39	83.33	87.55
MoEAD [62]	✓	93.96	95.77	92.38	98.33	98.87	98.13	91.92	98.81	89.81	94.51	**98.30**	94.93	91.42	88.89	90.02	96.98	89.84	88.62
LTOAD	✓	**94.94**	95.51	**94.43**	97.90	98.55	96.47	91.17	98.81	94.10	91.56	98.05	93.44	**96.08**	**95.17**	**93.53**	95.88	**92.85**	90.47

Table A14. **Comparison (↑) on MVTec** *exp200* [34] in pixel-level AUROC using the same evaluation metrics as Tab. A12.

Method	CA	$\mathcal{C}$	$\mathcal{C}^{\text{head}}$	$\mathcal{C}^{\text{tail}}$	*hazelnut*	*leather*	*bottle*	*wood*	*carpet*	*tile*	*metal nut*	*toothbrush*	*zipper*	*transistor*	*grid*	*pill*	*capsule*	*cable*	*screw*
MKD [75]	✗	79.61	84.34	75.46	96.61	78.43	98.81	91.32	75.96	85.14	64.13	76.39	90.36	79.33	62.91	78.18	69.57	78.86	68.15
DRAEM [91]	✗	69.82	84.00	57.41	**99.96**	95.78	99.44	**100.00**	97.47	99.67	94.72	72.77	70.79	51.12	28.15	57.50	37.89	50.50	**90.57**
RegAD [36]	✗	81.54	96.34	68.59	90.53	**100.00**	99.76	98.50	95.94	95.90	93.76	70.27	74.69	83.70	66.41	68.97	64.33	63.05	57.31
AnomalyGPT [25]	✗	85.95	98.69	74.79	98.93	**100.00**	97.94	95.96	99.60	99.39	99.07	88.06	86.19	70.69	92.65	86.82	61.47	51.20	61.28
LTAD [34]	✗	87.36	99.33	76.89	99.79	99.93	99.76	99.12	99.96	99.17	97.61	78.89	86.50	87.17	90.48	74.96	66.61	76.31	54.21
HVQ [57]	✗	82.14	99.30	67.12	99.79	**100.00**	99.76	97.54	99.60	**99.86**	98.53	83.33	80.86	69.00	81.04	68.60	44.52	66.12	43.49
UniAD [89]	✓	83.37	**99.52**	69.24	99.92	**100.00**	**100.00**	98.07	99.67	99.81	99.21	80.55	91.46	57.75	80.45	66.50	61.38	58.30	57.55
MoEAD [62]	✓	84.40	98.79	71.82	99.89	**100.00**	99.52	97.28	99.92	96.50	98.39	92.78	84.48	74.54	71.85	80.69	63.90	57.25	49.07
LTOAD	✓	**92.33**	99.45	**86.11**	**99.96**	**100.00**	99.60	98.16	**100.00**	98.88	**99.56**	**94.17**	**91.89**	**91.83**	**97.58**	**92.25**	**80.97**	**84.75**	55.42

Table A15. **Comparison (↑) on MVTec** *step100* [34] in image-level AUROC using the same evaluation metrics as Tab. A11.

Method	CA	$\mathcal{C}$	$\mathcal{C}^{\text{head}}$	$\mathcal{C}^{\text{tail}}$	*hazelnut*	*leather*	*bottle*	*wood*	*carpet*	*tile*	*metal nut*	*toothbrush*	*zipper*	*transistor*	*grid*	*pill*	*capsule*	*cable*	*screw*
MKD [75]	✗	85.90	86.92	85.01	94.61	96.24	92.63	77.85	93.20	76.96	77.01	93.87	90.92	73.73	72.96	89.09	93.41	72.46	93.60
DRAEM [91]	✗	79.65	92.81	69.79	94.54	97.87	89.75	95.11	98.08	**98.86**	80.53	90.82	74.91	60.30	50.84	76.96	65.39	53.87	85.26
RegAD [36]	✗	95.10	**97.25**	93.21	98.24	99.22	97.92	**95.87**	98.37	94.37	**96.78**	95.69	**95.71**	93.16	79.73	**96.53**	95.91	**94.54**	**94.48**
AnomalyGPT [25]	✗	89.28	93.62	85.49	88.41	**99.56**	94.05	89.61	**99.08**	95.52	89.09	95.53	92.65	65.87	94.42	79.26	86.48	83.11	86.62
LTAD [34]	✗	93.83	95.90	92.01	98.24	99.01	95.69	91.46	98.68	91.82	96.43	96.71	90.17	93.68	91.90	91.10	95.73	91.80	85.02
HVQ [57]	✗	94.17	96.56	92.08	**98.77**	98.98	**98.63**	93.35	98.89	92.84	94.47	98.09	90.76	87.04	92.77	91.26	94.89	91.97	89.86
UniAD [89]	✓	91.47	96.15	87.38	97.92	98.54	97.83	93.48	98.38	92.37	94.57	95.59	93.79	75.76	84.46	82.02	94.43	81.96	91.04
MoEAD [62]	✓	93.76	95.66	92.10	97.63	98.83	98.09	91.51	98.78	89.87	94.88	**98.25**	92.46	93.63	87.05	90.62	**96.99**	90.77	87.00
LTOAD	✓	**95.11**	95.68	**94.62**	97.97	98.63	96.65	91.11	98.77	93.97	92.63	97.87	92.54	**96.32**	**94.60**	93.17	96.08	93.95	92.39

Table A16. **Comparison (↑) on MVTec** *step100* [34] in pixel-level AUROC using the same evaluation metrics as Tab. A12.

Method	CA	$\mathcal{C}$	$\mathcal{C}^{\text{head}}$	$\mathcal{C}^{\text{tail}}$	*hazelnut*	*leather*	*bottle*	*wood*	*carpet*	*tile*	*metal nut*	*toothbrush*	*zipper*	*transistor*	*grid*	*pill*	*capsule*	*cable*	*screw*
MKD [75]	✗	79.31	84.48	74.79	97.04	80.03	99.05	91.58	75.52	85.21	62.90	72.78	**90.70**	77.67	64.75	76.38	**69.45**	78.94	67.70
DRAEM [91]	✗	71.64	90.90	54.79	96.96	96.39	86.50	98.24	94.98	97.58	65.68	64.44	46.90	39.29	57.81	63.39	40.76	51.55	**74.21**
AnomalyGPT [25]	✗	82.47	98.07	68.82	98.82	**100.00**	97.30	92.81	99.80	98.45	99.32	73.89	88.46	64.00	92.77	84.79	48.64	50.67	47.39
LTAD [34]	✗	85.60	99.15	73.74	99.43	99.93	99.68	**99.39**	99.80	99.21	96.58	75.00	85.90	81.71	90.48	76.68	63.98	65.54	50.65
HVQ [57]	✗	83.00	99.34	68.70	99.96	**100.00**	**100.00**	97.02	99.40	99.24	99.76	**93.33**	86.87	72.21	68.50	75.10	52.61	54.55	46.38
UniAD [89]	✓	81.32	**99.61**	65.31	**100.00**	**100.00**	**100.00**	98.15	99.39	**99.92**	**99.85**	81.11	82.32	54.58	86.38	53.19	64.22	50.52	50.21
MoEAD [62]	✓	83.13	98.96	69.27	99.61	**100.00**	99.52	97.46	99.56	97.58	99.02	90.83	87.79	73.29	69.76	76.68	54.13	53.82	47.90
LTOAD	✓	**88.62**	99.53	**79.07**	**100.00**	**100.00**	99.76	98.60	**100.00**	98.81	99.56	92.50	75.08	**83.21**	**95.91**	**88.82**	65.34	**89.22**	42.47

Table A17. **Comparison (↑) on MVTec** *step200* [34] in image-level AUROC using the same evaluation metrics as Tab. A11.

Method	CA	$\mathcal{C}$	$\mathcal{C}^{\text{head}}$	$\mathcal{C}^{\text{tail}}$	*hazelnut*	*leather*	*bottle*	*wood*	*carpet*	*tile*	*metal nut*	*toothbrush*	*zipper*	*transistor*	*grid*	*pill*	*capsule*	*cable*	*screw*
MKD [75]	✗	86.03	87.22	84.99	94.80	96.73	93.34	78.61	93.33	76.63	77.16	93.93	90.73	73.95	74.24	88.43	93.35	71.44	**93.86**
DRAEM [91]	✗	76.79	84.23	70.29	95.21	88.39	76.81	82.64	94.62	89.27	62.70	89.65	70.96	56.29	57.39	85.65	54.66	56.30	91.44
AnomalyGPT [25]	✗	89.45	93.75	85.69	88.29	**99.56**	95.12	89.94	**99.00**	**95.49**	88.82	94.98	91.61	66.40	**93.55**	78.70	90.56	82.42	87.33
LTAD [34]	✗	92.12	95.45	89.20	97.83	99.06	95.76	91.00	98.72	90.59	95.22	96.45	89.00	84.38	89.24	89.94	92.37	84.92	87.32
HVQ [57]	✗	92.61	**96.42**	89.28	**98.66**	98.86	**98.24**	92.21	98.58	91.75	**96.62**	98.31	91.54	83.37	84.53	88.15	95.09	88.13	85.11
UniAD [89]	✓	89.29	96.10	83.34	97.93	98.57	97.83	**93.46**	98.36	92.25	94.32	92.69	90.73	66.60	76.28	78.25	92.27	79.87	90.05
MoEAD [62]	✓	92.76	95.85	90.05	98.28	98.87	98.17	91.66	98.75	89.87	95.35	**98.32**	**91.72**	83.36	88.61	87.77	**95.39**	87.56	87.69
LTOAD	✓	**94.00**	95.93	**92.31**	98.09	98.64	96.72	91.26	98.74	93.88	94.15	97.62	90.66	**93.92**	92.76	**91.37**	90.84	**91.13**	90.17

Table A18. **Comparison (↑) on MVTec** *step200* [34] in pixel-level AUROC using the same evaluation metrics as Tab. A12.

Method	CA	$\mathcal{C}$	$\mathcal{C}^{\text{head}}$	$\mathcal{C}^{\text{tail}}$	*PCB3*	*PCB2*	*PCB1*	*PCB4*	*macaroni1*	*macaroni2*	*capsule*	*candle*	*fryum*	*capsules*	*chewinggum*	*pipe fryum*
RegAD [36]	✗	71.36	66.39	76.34	62.71	63.79	73.89	82.82	62.37	52.79	76.12	71.99	73.82	56.07	92.92	87.13
AnomalyGPT [25]	✗	70.34	70.13	70.56	62.69	78.55	79.57	69.66	**78.76**	51.54	49.96	77.15	80.77	63.88	71.76	79.84
LTAD [34]	✗	80.00	76.29	83.71	82.18	81.42	85.02	93.63	56.46	59.02	84.46	82.46	85.70	69.37	96.16	84.10
HVQ [57]	✗	78.63	**81.75**	75.51	**83.03**	**92.40**	**85.08**	**98.36**	75.28	56.36	**93.38**	61.10	79.50	63.77	86.64	68.70
UniAD [89]	✓	77.31	77.42	77.19	78.34	84.69	82.02	95.96	70.68	52.88	87.10	81.44	68.60	52.75	95.54	77.74
MoEAD [62]	✓	73.89	72.37	75.42	71.75	76.87	84.26	86.87	59.18	55.27	89.08	71.06	76.40	70.17	81.72	64.08
LTOAD	✓	**85.26**	79.64	**90.87**	80.84	81.15	84.28	96.42	70.71	**64.43**	91.17	**92.96**	**86.88**	**79.07**	**97.98**	**97.18**

Table A19. **Comparison (↑) on VisA** *exp100* [34] in image-level AUROC using the same evaluation metrics as Tab A11.

Method	CA	$\mathcal{C}$	$\mathcal{C}^{\text{head}}$	$\mathcal{C}^{\text{tail}}$	*PCB3*	*PCB2*	*PCB1*	*PCB4*	*macaroni1*	*macaroni2*	*capsule*	*candle*	*fryum*	*capsules*	*chewinggum*	*pipe fryum*
RegAD [36]	✗	94.40	93.72	95.07	95.27	92.58	**97.26**	91.82	94.88	90.53	95.03	96.82	94.97	87.20	97.37	99.07
AnomalyGPT [25]	✗	80.32	77.43	83.20	79.41	84.80	77.09	82.17	70.68	70.44	77.41	86.70	90.21	62.72	92.10	90.07
LTAD [34]	✗	95.56	95.16	95.97	**97.62**	96.06	92.18	95.96	96.58	92.58	97.90	99.08	95.39	85.77	98.82	98.83
HVQ [57]	✗	96.18	**95.99**	96.37	97.52	**97.21**	91.59	96.65	**98.01**	94.96	**98.54**	98.16	96.14	90.66	95.68	99.05
UniAD [89]	✓	95.03	94.57	95.48	97.05	96.12	91.61	**96.80**	97.04	88.84	98.19	98.82	95.25	83.21	98.68	98.78
MoEAD [62]	✓	94.34	93.36	95.31	95.73	96.25	91.52	94.98	92.93	88.72	97.53	98.98	95.10	83.05	97.86	**99.37**
LTOAD	✓	**96.91**	95.84	**97.97**	97.43	96.45	93.37	96.50	96.16	**95.13**	98.40	**99.16**	**96.67**	**95.40**	**98.87**	99.33

Table A20. **Comparison (↑) on VisA** *exp100* [34] in pixel-level AUROC using the same evaluation metrics as Tab A12.

Method	CA	$\mathcal{C}$	$\mathcal{C}^{\text{head}}$	$\mathcal{C}^{\text{tail}}$	*PCB3*	*PCB2*	*PCB1*	*PCB4*	*macaroni1*	*macaroni2*	*capsule*	*candle*	*fryum*	*capsules*	*chewinggum*	*pipe fryum*
RegAD [36]	✗	72.10	67.19	77.02	61.92	64.41	73.16	79.95	67.55	56.16	76.53	77.21	79.73	55.81	88.92	83.94
AnomalyGPT [25]	✗	69.78	69.22	70.35	61.64	75.82	79.42	68.41	**79.60**	50.40	45.90	76.59	79.80	63.34	72.83	83.63
LTAD [34]	✗	80.21	76.81	83.60	**84.98**	83.77	**86.86**	95.37	57.66	52.20	85.73	81.82	83.30	63.23	96.82	90.72
HVQ [57]	✗	78.03	79.35	76.71	78.50	**88.97**	82.56	**98.07**	64.13	**63.84**	**92.95**	63.76	83.30	63.85	87.04	69.36
UniAD [89]	✓	76.87	76.25	77.49	78.91	84.96	81.85	96.34	62.01	53.44	86.91	78.60	74.84	54.08	94.28	76.28
MoEAD [62]	✓	73.58	73.36	73.80	74.45	77.77	84.22	91.63	55.66	56.44	88.07	73.94	79.30	65.67	71.76	64.06
LTOAD	✓	**85.24**	**80.14**	**90.34**	80.46	82.09	84.80	97.03	74.46	61.99	91.85	**93.10**	**85.04**	**75.52**	**98.74**	**97.82**

Table A21. **Comparison (↑) on VisA** *exp200* [34] in image-level AUROC using the same evaluation metrics as Tab A11.

Method	CA	$\mathcal{C}$	$\mathcal{C}^{\text{head}}$	$\mathcal{C}^{\text{tail}}$	*PCB3*	*PCB2*	*PCB1*	*PCB4*	*macaroni1*	*macaroni2*	*capsule*	*candle*	*fryum*	*capsules*	*chewinggum*	*pipe fryum*
RegAD [36]	✗	94.69	94.07	95.30	95.78	94.22	**97.87**	90.41	96.18	90.01	95.64	96.80	95.89	87.54	96.73	99.23
AnomalyGPT [25]	✗	79.48	76.64	82.32	77.59	83.31	76.89	81.92	68.29	71.84	77.66	83.39	89.57	61.44	92.37	89.46
LTAD [34]	✗	95.36	94.98	95.69	**97.66**	95.03	91.56	96.42	96.20	89.58	98.31	99.02	95.16	83.74	98.87	99.02
HVQ [57]	✗	95.32	95.30	95.35	96.94	95.71	92.34	96.20	**97.19**	93.40	98.20	97.64	95.50	86.60	95.27	98.88
UniAD [89]	✓	94.80	94.34	95.26	97.07	96.06	91.64	96.53	96.18	88.56	98.06	98.61	95.24	82.72	98.32	98.66
MoEAD [62]	✓	93.98	93.52	94.44	95.91	96.51	91.89	95.40	92.45	88.95	97.78	98.92	95.62	79.04	96.41	98.89
LTOAD	✓	**96.97**	**96.08**	**97.86**	97.57	**96.64**	93.88	**96.93**	96.22	**95.26**	**98.45**	**99.30**	**96.52**	**94.75**	**98.89**	**99.26**

Table A22. **Comparison (↑) on VisA** *exp200* [34] in pixel-level AUROC using the same evaluation metrics as Tab A12.

Method	CA	$\mathcal{C}$	$\mathcal{C}^{\text{head}}$	$\mathcal{C}^{\text{tail}}$	*PCB3*	*PCB2*	*PCB1*	*PCB4*	*macaroni1*	*macaroni2*	*capsule*	*candle*	*fryum*	*capsules*	*chewinggum*	*pipe fryum*
AnomalyGPT [25]	✗	68.18	68.35	68.00	60.83	76.03	78.94	66.37	**77.50**	50.42	51.00	74.71	72.72	61.73	73.85	74.00
LTAD [34]	✗	78.53	76.56	80.50	**84.22**	83.85	**87.05**	94.55	50.11	59.56	81.07	77.84	77.36	66.48	94.24	85.98
HVQ [57]	✗	74.96	73.89	76.04	70.34	79.46	85.41	93.28	58.97	55.87	**89.86**	73.38	77.96	59.58	90.20	65.24
UniAD [89]	✓	73.67	76.61	70.73	79.17	**85.11**	82.97	96.38	62.94	53.13	85.70	65.38	76.92	53.11	89.22	54.06
MoEAD [62]	✓	70.93	72.68	69.18	73.67	75.50	83.28	85.91	58.69	59.04	84.23	66.08	77.70	63.85	70.22	52.98
LTOAD	✓	**83.89**	**80.01**	**87.77**	81.59	82.56	85.75	**96.76**	71.49	**61.93**	89.14	**91.52**	**87.04**	**70.70**	**96.90**	**91.34**

Table A23. **Comparison (↑) on VisA** *exp500* [34] in image-level AUROC using the same evaluation metrics as Tab A11.

Method	CA	$\mathcal{C}$	$\mathcal{C}^{\text{head}}$	$\mathcal{C}^{\text{tail}}$	*PCB3*	*PCB2*	*PCB1*	*PCB4*	*macaroni1*	*macaroni2*	*capsule*	*candle*	*fryum*	*capsules*	*chewinggum*	*pipe fryum*
AnomalyGPT [25]	✗	78.83	76.04	81.62	78.63	84.05	75.66	83.38	65.46	69.04	77.46	81.01	89.62	58.23	93.03	90.36
LTAD [34]	✗	94.66	94.42	94.90	**97.75**	95.60	92.76	96.23	94.80	89.39	97.54	98.86	95.16	80.98	98.69	98.18
HVQ [57]	✗	95.77	95.26	96.28	96.46	96.31	**94.94**	95.69	94.87	93.30	**98.17**	98.53	**96.10**	89.62	97.44	97.81
UniAD [89]	✓	94.35	94.35	94.35	97.01	96.21	91.49	96.56	**95.92**	88.95	97.96	98.35	94.94	81.48	97.76	95.62
MoEAD [62]	✓	93.39	93.06	93.71	95.73	96.15	91.40	94.81	91.14	89.16	97.40	98.68	94.93	79.83	94.72	96.71
LTOAD	✓	**96.60**	**95.78**	**97.41**	97.42	**96.39**	93.17	**96.81**	95.86	**95.05**	98.16	**99.09**	96.06	**93.42**	**98.82**	**98.91**

Table A24. **Comparison (↑) on VisA** *exp500* [34] in pixel-level AUROC using the same evaluation metrics as Tab A12.

Method	CA	$\mathcal{C}$	$\mathcal{C}^{\text{head}}$	$\mathcal{C}^{\text{tail}}$	*PCB3*	*PCB2*	*PCB1*	*PCB4*	*macaroni1*	*macaroni2*	*capsule*	*candle*	*fryum*	*capsules*	*chewinggum*	*pipe fryum*
RegAD [36]	✗	71.80	66.71	76.89	61.04	65.64	72.92	81.27	66.91	52.48	74.35	73.46	71.42	57.14	91.51	93.49
AnomalyGPT [25]	✗	71.89	71.41	72.37	65.90	75.90	74.12	75.06	82.88	54.61	50.73	80.22	**94.65**	57.10	76.24	75.26
LTAD [34]	✗	84.80	85.98	83.63	85.21	82.08	87.91	96.97	**86.17**	77.54	81.94	83.94	78.78	69.55	97.10	90.44
HVQ [57]	✗	76.68	87.24	66.12	**85.43**	**91.02**	92.17	99.02	83.15	72.63	64.42	53.44	76.22	58.80	73.76	70.06
UniAD [89]	✓	78.83	83.06	74.61	79.91	85.51	88.36	97.41	82.31	64.89	77.53	60.90	77.34	56.80	93.44	81.66
MoEAD [62]	✓	74.12	86.58	61.67	81.54	88.64	**92.59**	99.03	84.86	72.80	62.97	55.96	65.96	48.30	74.72	62.10
LTOAD	✓	**88.32**	**87.39**	**89.25**	85.21	87.04	90.80	**99.20**	83.46	**78.63**	**84.99**	**91.56**	86.88	**75.92**	**98.72**	**97.44**

Table A25. **Comparison (↑) on VisA** *step100* [34] in image-level AUROC using the same evaluation metrics as Tab A11.

Method	CA	$\mathcal{C}$	$\mathcal{C}^{\text{head}}$	$\mathcal{C}^{\text{tail}}$	*PCB3*	*PCB2*	*PCB1*	*PCB4*	*macaroni1*	*macaroni2*	*capsule*	*candle*	*fryum*	*capsules*	*chewinggum*	*pipe fryum*
RegAD [36]	✗	94.99	94.81	95.16	95.09	94.76	97.76	93.17	95.63	92.49	95.72	97.49	95.23	84.83	98.61	99.10
AnomalyGPT [25]	✗	82.30	79.41	85.19	80.35	86.10	78.21	80.42	76.54	74.83	81.56	91.79	91.02	60.06	93.07	93.63
LTAD [34]	✗	96.57	97.43	95.72	97.67	96.10	98.80	97.02	98.10	96.87	**96.57**	98.97	95.78	84.87	**98.93**	99.20
HVQ [57]	✗	95.63	97.69	93.57	97.56	**97.44**	98.58	96.79	**98.40**	97.40	89.61	97.35	94.26	87.80	93.57	98.84
UniAD [89]	✓	96.04	97.37	94.71	97.25	96.91	**99.18**	97.42	98.21	95.30	95.89	96.89	94.24	83.91	98.51	98.84
MoEAD [62]	✓	94.37	97.51	91.23	97.54	97.29	99.04	97.04	**98.40**	95.75	89.52	96.43	93.91	75.33	93.66	98.55
LTOAD	✓	**97.54**	**97.78**	**97.30**	**97.74**	96.89	99.11	**97.76**	97.48	**97.67**	96.23	**99.11**	**96.10**	**94.28**	98.82	**99.25**

Table A26. **Comparison (↑) on VisA** *step100* [34] in pixel-level AUROC using the same evaluation metrics as Tab A12.

Method	CA	$\mathcal{C}$	$\mathcal{C}^{\text{head}}$	$\mathcal{C}^{\text{tail}}$	*PCB3*	*PCB2*	*PCB1*	*PCB4*	*macaroni1*	*macaroni2*	*capsule*	*candle*	*fryum*	*capsules*	*chewinggum*	*pipe fryum*
RegAD [36]	✗	71.65	67.02	76.28	57.60	65.20	72.95	81.89	67.74	56.76	72.69	72.51	73.50	56.74	89.39	92.86
AnomalyGPT [25]	✗	69.78	71.19	68.38	66.79	75.20	74.30	73.57	81.94	55.34	44.74	82.28	74.26	56.81	77.68	74.49
LTAD [34]	✗	84.03	87.34	80.72	85.00	82.47	90.49	96.12	**88.64**	**81.33**	81.65	84.50	71.84	61.10	95.40	89.82
HVQ [57]	✗	75.52	**88.66**	62.38	**86.09**	**92.04**	92.99	**99.32**	84.42	77.09	50.99	63.18	69.14	47.13	79.34	64.52
UniAD [89]	✓	77.64	84.51	70.78	81.51	86.73	89.78	97.83	83.51	67.72	64.57	55.58	73.70	56.20	91.30	83.36
MoEAD [62]	✓	76.77	87.84	65.71	82.88	88.74	**93.24**	99.25	84.96	77.96	69.15	55.90	65.86	58.28	77.46	67.58
LTOAD	✓	**87.04**	87.77	**86.31**	85.80	86.74	91.24	99.03	84.68	79.11	**83.98**	**88.78**	**83.18**	**67.02**	**97.38**	**97.50**

Table A27. **Comparison (↑) on VisA** *step200* [34] in image-level AUROC using the same evaluation metrics as Tab A11.

Method	CA	$\mathcal{C}$	$\mathcal{C}^{\text{head}}$	$\mathcal{C}^{\text{tail}}$	*PCB3*	*PCB2*	*PCB1*	*PCB4*	*macaroni1*	*macaroni2*	*capsule*	*candle*	*fryum*	*capsules*	*chewinggum*	*pipe fryum*
RegAD [36]	✗	94.52	93.73	95.32	94.80	93.28	97.06	90.95	94.87	91.44	95.24	98.00	95.30	87.24	97.00	99.14
AnomalyGPT [25]	✗	81.97	78.69	85.25	80.22	85.88	77.08	81.05	74.01	73.90	80.96	91.60	90.77	61.16	93.04	93.99
LTAD [34]	✗	96.27	97.63	94.92	97.78	96.36	98.89	96.90	98.45	97.39	95.87	98.65	95.23	81.87	**98.82**	99.06
HVQ [57]	✗	95.50	**97.89**	93.10	97.73	97.50	**99.34**	97.00	98.19	**97.57**	90.31	96.45	95.05	82.89	95.17	98.76
UniAD [89]	✓	95.66	97.45	93.87	97.41	96.87	99.15	97.53	98.38	95.40	95.37	95.29	93.63	82.25	98.10	98.61
MoEAD [62]	✓	94.82	97.80	91.84	97.77	**97.60**	99.22	96.94	**98.64**	96.65	90.07	96.37	95.28	77.05	93.76	98.49
LTOAD	✓	**97.23**	97.74	**96.71**	**97.79**	96.89	99.03	**97.64**	97.73	97.37	**96.25**	**98.88**	**96.03**	**91.12**	98.74	**99.20**

Table A28. **Comparison (↑) on VisA** *step200* [34] in pixel-level AUROC using the same evaluation metrics as Tab A12.

Method	CA	$\mathcal{C}$	$\mathcal{C}^{\text{head}}$	$\mathcal{C}^{\text{tail}}$	*PCB3*	*PCB2*	*PCB1*	*PCB4*	*macaroni1*	*macaroni2*	*capsule*	*candle*	*fryum*	*capsules*	*chewinggum*	*pipe fryum*
AnomalyGPT [25]	✗	62.88	70.15	55.61	63.36	74.22	74.40	75.84	79.34	53.74	41.40	71.25	66.18	49.11	41.74	63.95
LTAD [34]	✗	83.33	88.40	78.26	85.68	85.19	89.95	96.68	87.91	**85.00**	78.59	76.56	69.86	**68.30**	**94.34**	81.88
HVQ [57]	✗	72.09	83.60	60.57	76.29	87.24	86.35	98.62	82.87	70.26	68.64	48.32	66.22	53.28	70.96	55.98
UniAD [89]	✓	71.84	81.84	61.85	78.79	84.72	88.80	97.44	79.46	61.83	61.82	68.54	55.96	51.30	68.28	65.22
MoEAD [62]	✓	72.75	**89.41**	56.09	83.37	**90.37**	**93.80**	99.03	**88.14**	81.77	67.31	38.26	57.20	54.97	64.60	54.18
LTOAD	✓	**84.84**	88.13	**81.56**	**85.99**	87.51	91.67	**99.29**	85.84	78.47	**87.26**	**78.94**	**76.72**	61.03	93.16	**92.26**

Table A29. **Comparison (↑) on VisA** *step500* [34] in image-level AUROC using the same evaluation metrics as Tab A11.

Method	CA	$\mathcal{C}$	$\mathcal{C}^{\text{head}}$	$\mathcal{C}^{\text{tail}}$	*PCB3*	*PCB2*	*PCB1*	*PCB4*	*macaroni1*	*macaroni2*	*capsule*	*candle*	*fryum*	*capsules*	*chewinggum*	*pipe fryum*
AnomalyGPT [25]	✗	81.48	78.88	84.09	79.88	84.91	78.51	79.59	75.69	74.69	78.52	92.32	90.94	59.75	89.52	93.49
LTAD [34]	✗	**96.41**	97.69	**95.12**	97.91	96.48	98.95	96.63	98.69	**97.50**	95.81	97.90	95.10	84.46	**98.52**	**98.93**
HVQ [57]	✗	95.06	97.34	92.77	97.06	97.15	99.03	96.87	97.90	96.05	91.24	96.33	**95.54**	82.58	93.91	97.03
UniAD [89]	✓	95.06	97.21	92.91	97.15	96.79	**99.16**	97.35	97.98	94.87	93.25	91.47	90.63	**90.63**	96.35	95.16
MoEAD [62]	✓	94.30	**98.10**	90.51	**97.95**	**97.78**	**99.16**	97.57	**98.96**	97.16	90.07	94.81	94.26	74.40	93.55	95.99
LTOAD	✓	**96.41**	97.74	95.08	97.63	96.90	99.07	**97.68**	97.71	97.44	**96.94**	**98.01**	94.39	84.83	98.34	97.96

Table A30. **Comparison (↑) on VisA** *step500* [34] in pixel-level AUROC using the same evaluation metrics as Tab A12.

Method	CA	$\mathcal{C}$	$\mathcal{C}^{\text{head}}$	$\mathcal{C}^{\text{tail}}$	*Class10*	*Class7*	*Class9*	*Class8*	*Class2*	*Class3*	*Class5*	*Class1*	*Class4*	*Class6*
RegAD [36]	✗	84.86	86.88	82.84	94.45	91.34	99.23	50.94	98.44	97.95	72.07	48.76	99.75	95.66
AnomalyGPT [25]	✗	84.86	84.68	85.04	82.55	**100.00**	87.28	70.07	83.52	93.61	96.88	83.91	53.00	97.79
LTAD [34]	✗	94.82	93.39	96.25	97.71	99.53	**99.91**	69.81	**100.00**	**99.69**	97.68	84.08	**100.00**	99.82
HVQ [57]	✗	76.65	76.78	76.52	95.41	74.27	66.92	50.13	97.17	61.06	73.82	**99.42**	94.66	53.64
UniAD [89]	✓	84.51	87.34	81.69	99.25	95.29	91.61	52.97	97.61	79.73	77.55	51.19	**100.00**	**99.99**
MoEAD [62]	✓	77.03	75.23	78.84	91.06	59.31	78.14	52.51	95.12	76.91	81.30	96.95	72.25	66.77
LTOAD	✓	**96.00**	**94.16**	**97.83**	**99.85**	99.90	99.86	**71.21**	**100.00**	99.32	**98.17**	92.47	**100.00**	99.21

Table A31. **Comparison (↑) on DAGM** *exp50* [34] in image-level AUROC using the same evaluation metrics as Tab. A11.

Method	CA	$\mathcal{C}$	$\mathcal{C}^{\text{head}}$	$\mathcal{C}^{\text{tail}}$	*Class10*	*Class7*	*Class9*	*Class8*	*Class2*	*Class3*	*Class5*	*Class1*	*Class4*	*Class6*
RegAD [36]	✗	90.29	91.91	88.66	98.74	88.41	**99.95**	73.28	99.17	93.69	85.23	72.23	**99.73**	92.44
AnomalyGPT [25]	✗	77.37	78.89	75.84	73.04	86.30	88.76	66.11	80.26	78.17	78.85	79.04	54.37	88.79
LTAD [34]	✗	97.40	97.11	**97.68**	99.09	97.49	99.93	**89.18**	**99.87**	**98.94**	**98.79**	93.90	99.49	97.27
HVQ [57]	✗	86.03	84.32	87.75	98.81	73.45	95.16	55.29	98.85	83.15	85.11	92.47	98.86	79.17
UniAD [89]	✓	90.70	89.75	91.64	**99.76**	93.54	99.29	56.45	99.74	90.87	89.53	79.49	98.91	**99.44**
MoEAD [62]	✓	86.08	84.02	88.15	98.46	68.93	97.60	56.45	98.66	87.27	87.73	89.69	97.00	79.03
LTOAD	✓	**97.41**	**97.34**	97.48	99.48	**99.04**	99.80	88.51	99.86	98.04	98.76	**95.76**	99.45	95.38

Table A32. **Comparison (↑) on DAGM** *exp50* [34] in pixel-level AUROC using the same evaluation metrics as Tab. A12.

Method	CA	$\mathcal{C}$	$\mathcal{C}^{\text{head}}$	$\mathcal{C}^{\text{tail}}$	*Class10*	*Class7*	*Class9*	*Class8*	*Class2*	*Class3*	*Class5*	*Class1*	*Class4*	*Class6*
RegAD [36]	✗	84.86	86.88	82.84	94.45	91.34	99.23	50.94	98.44	97.95	72.07	48.76	99.75	95.66
AnomalyGPT [25]	✗	85.31	86.05	84.58	86.56	**100.00**	89.76	**71.90**	82.03	94.29	95.42	84.22	52.27	96.69
LTAD [34]	✗	94.40	93.24	95.56	97.78	99.60	**99.89**	68.92	**100.00**	**99.66**	97.35	81.01	**100.00**	99.78
HVQ [57]	✗	75.42	75.88	74.96	94.17	70.69	67.04	51.00	96.49	62.20	74.47	**99.59**	86.08	52.44
UniAD [89]	✓	84.34	87.66	81.02	99.15	96.01	92.45	53.31	97.40	79.79	74.26	51.15	99.91	**99.99**
MoEAD [62]	✓	73.26	72.53	73.99	88.93	56.55	73.94	50.73	92.50	68.31	77.81	97.08	61.74	65.02
LTOAD	✓	**95.58**	**93.75**	**97.41**	**99.60**	99.92	99.79	69.45	**100.00**	99.60	**97.66**	90.94	**100.00**	98.84

Table A33. **Comparison (↑) on DAGM** *exp100* [34] in image-level AUROC using the same evaluation metrics as Tab. A11.

Method	CA	$\mathcal{C}$	$\mathcal{C}^{\text{head}}$	$\mathcal{C}^{\text{tail}}$	*Class10*	*Class7*	*Class9*	*Class8*	*Class2*	*Class3*	*Class5*	*Class1*	*Class4*	*Class6*
RegAD [36]	✗	90.28	91.91	88.66	98.74	88.40	**99.95**	73.28	99.17	93.69	85.23	72.23	**99.73**	92.44
AnomalyGPT [25]	✗	77.20	79.19	75.21	74.34	86.14	89.99	66.26	79.22	77.37	77.79	80.49	54.93	85.45
LTAD [34]	✗	97.30	97.01	**97.58**	99.09	97.42	99.93	**88.74**	**99.87**	**98.95**	98.62	93.76	99.59	96.98
HVQ [57]	✗	85.54	84.07	87.01	98.82	71.83	95.48	55.51	98.70	83.71	84.81	92.75	98.34	75.44
UniAD [89]	✓	90.13	89.66	90.60	**99.75**	93.54	99.33	55.96	99.76	90.13	87.54	78.19	97.74	**99.42**
MoEAD [62]	✓	84.99	83.07	86.92	97.76	67.49	96.90	55.18	98.03	85.14	85.19	89.18	95.54	79.53
LTOAD	✓	**97.32**	**97.26**	97.39	99.41	**98.86**	99.79	88.37	99.86	98.05	**98.67**	**95.61**	99.41	95.20

Table A34. **Comparison (↑) on DAGM** *exp100* [34] in pixel-level AUROC using the same evaluation metrics as Tab. A12.

Method	CA	$\mathcal{C}$	$\mathcal{C}^{\text{head}}$	$\mathcal{C}^{\text{tail}}$	*Class10*	*Class7*	*Class9*	*Class8*	*Class2*	*Class3*	*Class5*	*Class1*	*Class4*	*Class6*
RegAD [36]	✗	84.86	86.88	82.84	94.45	91.34	99.23	50.94	98.44	97.95	72.07	48.76	99.75	95.66
AnomalyGPT [25]	✗	83.29	83.45	83.13	84.31	**100.00**	87.16	66.80	79.00	89.90	95.65	80.03	52.49	97.56
LTAD [34]	✗	94.29	93.08	95.51	97.78	99.62	99.80	68.21	**100.00**	99.51	97.34	80.90	**100.00**	99.82
HVQ [57]	✗	76.19	76.36	76.03	95.78	73.96	64.36	50.82	96.90	62.25	75.00	**99.57**	88.06	55.25
UniAD [89]	✓	83.56	87.09	80.02	99.40	95.42	89.82	53.32	97.53	79.13	72.48	50.52	97.98	**99.99**
MoEAD [62]	✓	72.61	72.11	73.10	89.16	54.77	73.66	51.42	91.55	67.92	77.01	96.98	61.91	61.69
LTOAD	✓	**94.71**	**93.79**	**95.63**	**99.51**	99.88	**99.89**	**69.66**	**100.00**	**99.57**	**98.27**	81.53	**100.00**	98.80

Table A35. **Comparison (↑) on DAGM** *exp200* [34] in image-level AUROC using the same evaluation metrics as Tab. A11.

Method	CA	$\mathcal{C}$	$\mathcal{C}^{\text{head}}$	$\mathcal{C}^{\text{tail}}$	*Class10*	*Class7*	*Class9*	*Class8*	*Class2*	*Class3*	*Class5*	*Class1*	*Class4*	*Class6*
RegAD [36]	✗	90.29	91.91	88.66	98.74	88.40	**99.95**	73.28	99.17	93.69	85.23	72.23	**99.73**	92.44
AnomalyGPT [25]	✗	77.16	79.58	74.73	74.85	85.83	89.38	65.89	81.93	73.30	77.54	81.12	56.02	85.69
LTAD [34]	✗	**97.19**	96.98	**97.39**	99.02	97.38	99.92	**88.72**	99.86	**98.88**	98.50	93.26	99.52	96.80
HVQ [57]	✗	85.32	84.22	86.42	98.89	72.99	94.89	55.59	98.75	84.66	85.63	91.97	98.11	71.70
UniAD [89]	✓	89.73	89.57	89.88	**99.77**	93.28	99.22	55.84	99.76	90.30	86.78	76.53	96.44	**99.39**
MoEAD [62]	✓	84.18	82.54	85.81	97.70	64.61	96.94	55.46	97.99	84.78	85.19	88.57	95.36	75.15
LTOAD	✓	97.11	**97.29**	96.94	99.41	**98.85**	99.81	88.48	**99.88**	98.01	**98.64**	**93.88**	99.41	94.76

Table A36. **Comparison (↑) on DAGM** *exp200* [34] in pixel-level AUROC using the same evaluation metrics as Tab. A12.

Method	CA	$\mathcal{C}$	$\mathcal{C}^{\text{head}}$	$\mathcal{C}^{\text{tail}}$	*Class10*	*Class7*	*Class9*	*Class8*	*Class2*	*Class3*	*Class5*	*Class1*	*Class4*	*Class6*
RegAD [36]	✗	84.86	86.88	82.84	94.45	91.34	99.23	50.94	98.44	97.95	72.07	48.76	99.75	95.66
AnomalyGPT [25]	✗	83.47	84.38	82.55	86.38	**99.99**	91.88	64.78	78.88	90.31	95.06	79.88	52.72	94.78
LTAD [34]	✗	93.54	93.23	**93.85**	97.76	99.77	99.79	68.81	**100.00**	99.36	97.04	73.00	**100.00**	99.83
HVQ [57]	✗	76.27	77.51	75.04	96.24	75.46	67.12	51.39	97.32	61.73	76.55	**99.06**	76.79	61.07
UniAD [89]	✓	81.35	88.75	73.96	**99.89**	98.39	96.60	51.31	97.56	85.24	75.57	53.49	55.64	**99.84**
MoEAD [62]	✓	72.28	72.81	71.74	89.39	57.43	72.99	52.05	92.20	68.71	75.06	94.66	59.14	61.14
LTOAD	✓	**93.82**	**93.98**	93.67	99.60	99.85	**99.82**	**70.62**	**100.00**	**99.51**	**97.67**	72.66	**100.00**	98.50

Table A37. **Comparison (↑) on DAGM** *exp500* [34] in image-level AUROC using the same evaluation metrics as Tab. A11.

Method	CA	$\mathcal{C}$	$\mathcal{C}^{\text{head}}$	$\mathcal{C}^{\text{tail}}$	*Class10*	*Class7*	*Class9*	*Class8*	*Class2*	*Class3*	*Class5*	*Class1*	*Class4*	*Class6*
RegAD [36]	✗	90.29	91.91	88.66	98.74	88.40	**99.95**	73.28	99.17	93.69	85.23	72.23	**99.73**	92.44
AnomalyGPT [25]	✗	76.87	78.93	74.81	76.11	85.42	90.89	62.51	79.73	72.28	77.29	82.21	57.35	84.93
LTAD [34]	✗	**97.01**	96.94	**97.07**	99.02	97.33	99.92	**88.57**	**99.87**	**98.74**	98.43	92.23	99.31	96.63
HVQ [57]	✗	85.41	84.74	86.08	98.94	74.08	95.27	56.52	98.89	83.16	85.12	89.74	96.55	75.87
UniAD [89]	✓	88.63	89.44	87.82	**99.77**	93.10	99.22	55.37	99.73	89.05	84.62	75.15	90.93	**99.33**
MoEAD [62]	✓	83.34	83.11	83.57	97.82	68.03	96.68	55.14	97.89	84.76	84.24	85.67	93.03	70.12
LTOAD	✓	96.98	**97.23**	96.72	99.39	**98.84**	99.83	88.25	99.86	98.10	**98.55**	**93.45**	99.13	94.30

Table A38. **Comparison (↑) on DAGM** *exp500* [34] in pixel-level AUROC using the same evaluation metrics as Tab. A12.

Method	CA	$\mathcal{C}$	$\mathcal{C}^{\text{head}}$	$\mathcal{C}^{\text{tail}}$	*Class6*	*Class4*	*Class1*	*Class5*	*Class3*	*Class2*	*Class8*	*Class9*	*Class7*	*Class10*
RegAD [36]	✗	84.85	82.83	86.88	95.66	99.75	48.76	72.07	97.95	98.44	50.94	99.23	91.34	94.45
AnomalyGPT [25]	✗	84.65	86.58	82.72	99.11	67.43	74.65	**97.48**	94.21	88.79	63.56	92.07	**99.96**	69.20
LTAD [34]	✗	94.09	**96.85**	91.34	99.76	**100.00**	88.00	97.19	99.29	**100.00**	67.80	99.75	95.03	94.12
HVQ [57]	✗	75.49	72.14	78.84	61.39	96.04	**99.65**	74.83	62.26	96.37	49.37	66.89	62.88	85.20
UniAD [89]	✓	82.48	85.78	79.17	**99.93**	**100.00**	63.41	79.04	86.52	97.00	51.42	98.31	71.48	77.67
MoEAD [62]	✓	71.11	66.07	76.15	66.77	69.09	96.90	78.52	69.49	91.20	47.45	62.87	49.29	79.56
LTOAD	✓	**95.24**	93.21	**97.27**	99.19	**100.00**	90.26	97.36	**99.54**	**100.00**	**68.78**	**99.89**	99.61	**97.78**

Table A39. **Comparison (↑) on DAGM** *reverse exp200* [34] in image-level AUROC using the same evaluation metrics as Tab. A11.

Method	CA	$\mathcal{C}$	$\mathcal{C}^{\text{head}}$	$\mathcal{C}^{\text{tail}}$	*Class6*	*Class4*	*Class1*	*Class5*	*Class3*	*Class2*	*Class8*	*Class9*	*Class7*	*Class10*
RegAD [36]	✗	90.28	88.66	91.91	92.44	99.73	72.23	85.23	93.69	99.17	73.28	**99.95**	88.40	98.74
AnomalyGPT [25]	✗	78.41	76.49	**8043.00**	94.92	57.21	69.48	81.27	79.10	89.15	65.79	95.83	83.85	67.53
LTAD [34]	✗	96.68	**97.63**	95.72	97.16	99.43	94.76	98.30	**98.52**	**99.87**	85.23	99.92	95.11	98.49
HVQ [57]	✗	85.58	82.49	88.66	83.07	98.97	92.35	85.48	83.45	98.71	56.00	94.12	66.84	96.78
UniAD [89]	✓	89.98	92.69	87.27	**99.49**	**99.86**	84.59	90.23	89.30	99.65	53.89	99.55	85.42	97.87
MoEAD [62]	✓	83.55	79.28	87.83	82.51	96.61	89.01	86.02	84.99	97.32	54.46	92.42	58.52	93.66
LTOAD	✓	**97.20**	96.97	97.43	95.92	99.33	**95.50**	**98.43**	97.96	**99.87**	**87.62**	99.83	**98.47**	**99.08**

Table A40. **Comparison (↑) on DAGM** *reverse exp200* [34] in pixel-level AUROC using the same evaluation metrics as Tab. A12.

Method	CA	$\mathcal{C}$	$\mathcal{C}^{\text{head}}$	$\mathcal{C}^{\text{tail}}$	*Class10*	*Class7*	*Class9*	*Class8*	*Class2*	*Class3*	*Class5*	*Class1*	*Class4*	*Class6*
RegAD [36]	✗	84.85	86.88	82.83	94.45	91.34	99.23	50.94	98.44	97.95	72.07	48.76	99.75	95.66
AnomalyGPT [25]	✗	87.27	88.29	86.24	89.51	**100.00**	88.81	70.88	92.25	95.97	96.69	83.77	57.59	97.20
LTAD [34]	✗	94.03	93.20	94.85	97.67	99.57	99.85	68.92	**99.99**	**99.38**	97.17	77.95	**100.00**	99.77
HVQ [57]	✗	76.87	76.49	77.25	95.39	72.25	64.81	52.44	97.56	62.58	75.96	**99.67**	94.61	53.42
UniAD [89]	✓	82.37	87.11	77.63	99.06	95.02	92.84	51.38	97.26	67.98	70.70	50.01	99.48	**99.99**
MoEAD [62]	✓	72.86	72.62	73.11	89.19	55.60	74.16	51.87	92.25	65.43	77.09	95.51	65.63	61.89
LTOAD	✓	**95.59**	**94.43**	**96.75**	**99.86**	99.93	**99.93**	**72.45**	**99.99**	99.36	**98.15**	86.84	**100.00**	99.42

Table A41. **Comparison (↑) on DAGM** *step50* [34] in image-level AUROC using the same evaluation metrics as Tab. A11.

Method	CA	$\mathcal{C}$	$\mathcal{C}^{\text{head}}$	$\mathcal{C}^{\text{tail}}$	*Class10*	*Class7*	*Class9*	*Class8*	*Class2*	*Class3*	*Class5*	*Class1*	*Class4*	*Class6*
RegAD [36]	✗	90.28	91.90	88.66	98.74	88.40	**99.95**	73.28	99.17	93.69	85.23	72.23	**99.73**	92.44
AnomalyGPT [25]	✗	78.28	79.78	76.77	75.75	86.67	89.54	66.73	80.23	79.02	79.42	81.74	55.34	88.31
LTAD [34]	✗	97.20	97.00	**97.39**	99.06	97.30	99.92	88.85	**99.86**	**98.92**	98.59	93.21	99.48	96.76
HVQ [57]	✗	85.56	83.98	87.14	98.87	71.92	94.41	55.80	98.88	82.24	85.39	92.53	98.78	76.78
UniAD [89]	✓	89.52	89.87	89.16	**99.71**	93.11	99.36	57.49	99.71	86.33	84.15	78.58	97.41	**99.37**
MoEAD [62]	✓	84.04	82.54	85.53	97.81	64.80	96.77	55.31	98.03	83.22	85.51	86.14	96.05	76.76
LTOAD	✓	**97.41**	**97.50**	97.32	99.53	**99.13**	99.81	**89.19**	99.85	98.02	**98.67**	**95.09**	99.45	95.39

Table A42. **Comparison (↑) on DAGM** *step50* [34] in pixel-level AUROC using the same evaluation metrics as Tab. A12.

Method	CA	$\mathcal{C}$	$\mathcal{C}^{\text{head}}$	$\mathcal{C}^{\text{tail}}$	*Class10*	*Class7*	*Class9*	*Class8*	*Class2*	*Class3*	*Class5*	*Class1*	*Class4*	*Class6*
RegAD [36]	✗	84.86	86.88	82.84	94.45	91.34	99.23	50.94	98.44	97.95	72.07	48.76	99.75	95.66
AnomalyGPT [25]	✗	86.48	88.35	84.60	90.24	**100.00**	87.79	70.42	93.31	95.21	96.41	81.82	51.01	98.56
LTAD [34]	✗	93.97	93.13	94.81	97.85	99.31	**99.87**	68.62	99.98	**99.28**	96.95	78.01	**100.00**	99.82
HVQ [57]	✗	75.78	77.71	73.85	96.14	77.21	66.00	51.44	97.76	59.91	70.10	**99.36**	84.07	55.83
UniAD [89]	✓	81.11	87.19	75.03	99.12	95.23	91.99	52.38	97.23	67.80	69.05	50.48	87.83	**99.99**
MoEAD [62]	✓	70.83	72.39	69.27	88.91	55.98	73.19	51.79	92.10	60.98	69.78	95.38	58.74	61.46
LTOAD	✓	**94.83**	**94.29**	**95.38**	**99.79**	99.96	99.86	**71.82**	**100.00**	99.10	**98.20**	80.93	**100.00**	98.65

Table A43. **Comparison (↑) on DAGM** *step100* [34] in image-level AUROC using the same evaluation metrics as Tab. A11.

Method	CA	$\mathcal{C}$	$\mathcal{C}^{\text{head}}$	$\mathcal{C}^{\text{tail}}$	*Class10*	*Class7*	*Class9*	*Class8*	*Class2*	*Class3*	*Class5*	*Class1*	*Class4*	*Class6*
RegAD [36]	✗	90.28	91.91	88.66	98.74	88.40	**99.95**	73.28	99.17	93.69	85.23	72.23	**99.73**	92.45
AnomalyGPT [25]	✗	78.76	80.23	77.29	77.01	86.46	89.08	67.51	81.07	79.40	80.00	83.19	55.72	88.14
LTAD [34]	✗	97.07	96.94	**97.20**	99.11	97.16	99.92	88.63	**99.86**	**98.51**	98.52	92.69	99.44	96.83
HVQ [57]	✗	85.23	84.53	85.92	99.00	74.70	94.23	55.83	98.91	81.82	83.28	91.50	97.72	75.26
UniAD [89]	✓	89.11	90.08	88.15	**99.71**	93.50	99.25	58.29	99.68	84.99	83.20	78.56	94.66	**99.34**
MoEAD [62]	✓	83.34	82.59	84.08	97.65	65.49	96.70	55.22	97.89	81.26	81.89	86.73	94.86	75.67
LTOAD	✓	**97.30**	**97.60**	97.00	99.51	**99.18**	99.82	**89.62**	**99.86**	97.96	**98.60**	**94.18**	99.49	94.79

Table A44. **Comparison (↑) on DAGM** *step100* [34] in pixel-level AUROC using the same evaluation metrics as Tab. A12.

Method	CA	$\mathcal{C}$	$\mathcal{C}^{\text{head}}$	$\mathcal{C}^{\text{tail}}$	*Class10*	*Class7*	*Class9*	*Class8*	*Class2*	*Class3*	*Class5*	*Class1*	*Class4*	*Class6*
RegAD [36]	✗	84.86	86.88	82.84	94.45	91.34	99.23	50.94	98.44	97.95	72.07	48.76	99.75	95.66
AnomalyGPT [25]	✗	84.73	87.34	82.12	89.58	**100.00**	86.36	70.28	90.49	95.43	96.29	79.86	41.35	97.66
LTAD [34]	✗	93.79	93.69	93.90	98.74	99.52	**99.91**	70.26	**100.00**	99.02	95.46	75.41	**100.00**	99.59
HVQ [57]	✗	75.92	77.47	74.36	96.59	79.49	61.29	52.59	97.40	60.29	71.22	**99.40**	78.90	61.98
UniAD [89]	✓	80.33	88.85	71.81	99.80	98.48	97.68	50.38	97.92	81.13	72.95	51.43	53.55	**99.97**
MoEAD [62]	✓	70.01	72.78	67.24	88.96	56.44	73.45	52.30	92.74	59.27	67.31	95.24	58.30	56.09
LTOAD	✓	**94.12**	**94.33**	**93.92**	**99.86**	99.98	**99.91**	**71.89**	**100.00**	**99.27**	**97.13**	74.53	**100.00**	98.66

Table A45. **Comparison (↑) on DAGM** *step200* [34] in image-level AUROC using the same evaluation metrics as Tab. A11.

Method	CA	$\mathcal{C}$	$\mathcal{C}^{\text{head}}$	$\mathcal{C}^{\text{tail}}$	*Class10*	*Class7*	*Class9*	*Class8*	*Class2*	*Class3*	*Class5*	*Class1*	*Class4*	*Class6*
RegAD [36]	✗	90.29	91.91	88.66	98.74	88.40	**99.95**	73.28	99.17	93.69	85.23	72.23	**99.73**	92.44
AnomalyGPT [25]	✗	78.29	79.40	77.19	75.97	86.68	89.22	65.88	79.26	80.28	79.87	82.95	55.26	87.57
LTAD [34]	✗	96.84	97.33	96.35	99.24	97.63	99.93	**89.98**	**99.89**	**98.12**	97.40	91.74	99.17	95.33
HVQ [57]	✗	84.63	84.15	85.11	99.05	74.48	92.43	55.92	98.87	80.50	81.55	90.88	97.08	75.56
UniAD [89]	✓	89.07	90.61	87.40	**99.72**	94.02	99.43	60.11	99.77	85.71	83.04	78.28	90.59	**99.40**
MoEAD [62]	✓	82.27	82.71	81.83	97.79	65.50	96.79	55.38	98.08	80.50	78.83	86.22	93.85	69.77
LTOAD	✓	**97.14**	**97.54**	**96.74**	99.53	**99.07**	99.81	89.40	99.86	97.77	**98.34**	**93.16**	99.38	95.03

Table A46. **Comparison (↑) on DAGM** *step200* [34] in pixel-level AUROC using the same evaluation metrics as Tab. A12.

Method	CA	$\mathcal{C}$	$\mathcal{C}^{\text{head}}$	$\mathcal{C}^{\text{tail}}$	*Class10*	*Class7*	*Class9*	*Class8*	*Class2*	*Class3*	*Class5*	*Class1*	*Class4*	*Class6*
RegAD [36]	✗	84.86	86.88	82.84	94.45	91.34	99.23	50.94	98.44	97.95	72.07	48.76	99.75	95.66
AnomalyGPT [25]	✗	85.08	88.73	81.43	91.35	**100.00**	89.07	71.74	91.48	93.30	**96.99**	77.94	41.77	97.14
LTAD [34]	✗	**92.78**	93.96	**91.59**	99.36	99.84	99.88	70.77	99.97	96.85	95.99	65.57	**100.00**	99.52
HVQ [57]	✗	73.86	76.98	70.73	96.83	76.83	62.44	51.30	97.50	61.45	67.09	**99.18**	63.12	62.83
UniAD [89]	✓	80.04	88.88	71.18	99.77	98.62	97.19	50.88	97.98	76.32	74.09	51.57	53.98	**99.98**
MoEAD [62]	✓	69.73	70.64	68.82	91.85	53.45	65.20	52.22	90.48	55.91	63.78	93.48	68.73	62.19
LTOAD	✓	92.57	**94.42**	90.72	**99.88**	99.94	**99.91**	**72.37**	**100.00**	**98.83**	95.85	61.96	**100.00**	96.94

Table A47. **Comparison (↑) on DAGM** *step500* [34] in image-level AUROC using the same evaluation metrics as Tab. A11.

Method	CA	$\mathcal{C}$	$\mathcal{C}^{\text{head}}$	$\mathcal{C}^{\text{tail}}$	*Class10*	*Class7*	*Class9*	*Class8*	*Class2*	*Class3*	*Class5*	*Class1*	*Class4*	*Class6*
RegAD [36]	✗	90.29	91.91	88.66	98.74	88.40	**99.95**	73.28	99.17	93.69	85.23	72.23	**99.73**	92.44
AnomalyGPT [25]	✗	78.75	79.93	77.57	77.90	86.27	89.89	65.79	79.78	78.81	80.68	84.82	56.79	86.73
LTAD [34]	✗	**96.65**	**97.67**	**95.64**	99.45	98.44	99.92	**90.66**	**99.87**	**97.07**	**97.88**	88.97	98.89	95.39
HVQ [57]	✗	84.11	84.28	83.95	99.01	74.41	92.91	56.10	98.97	79.25	80.49	89.51	94.71	75.76
UniAD [89]	✓	88.53	90.43	86.62	**99.73**	94.39	99.42	58.89	99.75	85.70	83.61	79.19	85.25	**99.39**
MoEAD [62]	✓	82.19	82.77	81.61	97.88	65.21	97.09	55.62	98.06	77.94	80.24	86.41	92.33	71.16
LTOAD	✓	96.52	97.50	95.54	99.52	**99.07**	99.81	89.23	99.86	97.01	97.87	**90.61**	99.05	93.18

Table A48. **Comparison (↑) on DAGM** *step500* [34] in pixel-level AUROC using the same evaluation metrics as Tab. A12.

Method	CA	$\mathcal{C}$	$\mathcal{C}^{\text{head}}$	$\mathcal{C}^{\text{tail}}$	*Class6*	*Class4*	*Class1*	*Class5*	*Class3*	*Class2*	*Class8*	*Class9*	*Class7*	*Class10*
RegAD [36]	✗	84.86	82.84	86.88	95.66	99.75	48.76	72.07	97.95	98.44	50.94	99.23	91.34	94.45
AnomalyGPT [25]	✗	83.64	88.06	79.21	98.71	66.42	80.31	**98.41**	96.47	70.21	64.49	91.11	**100.00**	70.22
LTAD [34]	✗	93.89	**97.40**	90.39	99.79	**100.00**	90.40	97.14	99.67	99.86	64.86	99.00	93.73	94.48
HVQ [57]	✗	74.51	69.85	79.17	61.73	92.70	**99.58**	76.33	65.52	87.01	52.87	63.89	58.16	87.31
UniAD [89]	✓	83.57	88.89	78.25	**99.98**	99.99	72.96	81.28	90.28	95.34	50.64	98.05	65.58	81.67
MoEAD [62]	✓	71.28	65.10	77.47	68.85	72.52	96.37	78.83	70.77	80.39	51.92	63.09	51.51	78.59
LTOAD	✓	**95.10**	92.50	**97.69**	99.33	**100.00**	91.43	97.92	**99.79**	**100.00**	**66.69**	**99.33**	99.50	**97.00**

Table A49. **Comparison (↑) on DAGM** *reverse step200* [34] in image-level AUROC using the same evaluation metrics as Tab. A11.

Method	CA	$\mathcal{C}$	$\mathcal{C}^{\text{head}}$	$\mathcal{C}^{\text{tail}}$	*Class6*	*Class4*	*Class1*	*Class5*	*Class3*	*Class2*	*Class8*	*Class9*	*Class7*	*Class10*
RegAD [36]	✗	90.28	88.67	91.91	92.44	99.73	72.23	85.24	93.69	99.17	73.28	**99.95**	88.41	**98.74**
AnomalyGPT [25]	✗	77.11	75.54	78.68	91.83	54.33	67.86	81.32	82.37	79.32	69.19	97.15	85.78	61.97
LTAD [34]	✗	96.66	**97.93**	95.40	97.22	99.49	95.31	98.57	**99.04**	99.79	84.36	99.89	94.52	98.45
HVQ [57]	✗	85.35	82.10	88.61	81.26	98.77	91.96	86.72	84.30	97.25	55.22	93.87	67.04	97.10
UniAD [89]	✓	90.79	94.62	86.96	**99.47**	**99.82**	87.93	93.47	92.41	99.28	54.11	99.48	83.94	98.00
MoEAD [62]	✓	83.48	79.00	87.96	83.47	96.74	87.69	86.56	85.33	94.50	55.44	91.26	60.05	93.76
LTOAD	✓	**96.96**	96.33	**97.58**	95.81	99.44	**95.90**	**98.66**	98.11	**99.83**	**85.41**	99.74	**97.96**	98.73

Table A50. **Comparison (↑) on DAGM** *reverse step200* [34] in pixel-level AUROC using the same evaluation metrics as Tab. A12.

Config.	Online	*B*	*B-HF*	*B-TF*	*D2-HF*	*D2-TF*	*D5-HF*	*D5-TF*	*D5-M*	Avg.
exp100	✗	95.08	95.08	95.08	95.08	95.08	95.08	95.08	95.08	95.08
	✓	**95.36**	**95.30**	**95.32**	**95.23**	**95.28**	**95.26**	**95.25**	**95.21**	**95.28**
exp200	✗	94.82	94.82	94.82	94.82	94.82	94.82	94.82	94.82	94.82
	✓	**95.05**	**95.01**	**95.10**	**95.01**	**95.01**	**94.89**	**94.94**	**94.91**	**94.99**
step100	✗	95.03	95.03	95.03	95.03	95.03	95.03	95.03	95.03	95.03
	✓	**95.26**	**95.26**	**95.28**	**95.17**	**95.12**	**95.04**	**95.14**	**95.09**	**95.17**
step200	✗	94.03	94.03	94.03	94.03	94.03	94.03	94.03	94.03	94.03
	✓	**94.83**	**94.72**	**94.85**	**94.60**	**94.82**	**94.55**	**94.67**	**94.53**	**94.70**

Table A51. **Comparison (↑) on online MVTec** in pixel-level AUROC for anomaly segmentation across 4 configurations [34] of $\mathcal{D}^{\intercal}$ and 8 configurations of $\mathcal{D}^{0}$, *i.e. B*, *B-HF*, *B-TF*, *D2-HF*, *D2-TF*, *D5-HF*, *D5-TF*, and *D5-M*.

Config.	Online	*B*	*B-HF*	*B-TF*	*D2-HF*	*D2-TF*	*D5-HF*	*D5-TF*	*D5-M*	Avg.
exp100	✗	97.05	97.05	97.05	97.05	97.05	97.05	97.05	97.05	97.05
	✓	**97.29**	**97.20**	**97.30**	**97.06**	**97.26**	**97.16**	**97.25**	**97.14**	**97.21**
exp200	✗	96.91	96.91	96.91	96.91	96.91	96.91	96.91	96.91	96.91
	✓	**97.09**	**96.98**	**97.14**	**97.02**	**97.05**	**97.03**	**96.97**	**97.09**	**97.05**
step100	✗	97.40	97.40	97.40	97.40	97.40	97.40	97.40	97.40	97.40
	✓	**97.61**	**97.54**	**97.65**	**97.52**	**97.55**	**97.45**	**97.50**	**97.53**	**97.54**
step200	✗	97.09	97.09	97.09	97.09	97.09	97.09	97.09	97.09	97.09
	✓	**97.49**	**97.43**	**97.49**	**97.17**	**97.39**	**97.19**	**97.30**	**97.29**	**97.34**

Table A52. **Comparison (↑) on online VisA** in the same format as Tab. A51.

Config.	Online	*B*	*B-HF*	*B-TF*	*D2-HF*	*D2-TF*	*D5-HF*	*D5-TF*	*D5-M*	Avg.
exp100	✗	**97.48**	97.48	97.48	**97.48**	**97.48**	**97.48**	**97.48**	**97.48**	**97.48**
	✓	97.47	**97.49**	**97.50**	97.42	97.47	97.42	97.38	97.30	97.43
exp200	✗	97.29	97.29	97.29	97.29	97.29	97.29	97.29	97.29	97.29
	✓	**97.44**	**97.44**	**97.44**	**97.40**	**97.43**	**97.40**	**97.38**	**97.03**	**97.37**
step100	✗	97.43	97.43	97.43	97.43	97.43	97.43	97.43	**97.43**	**97.43**
	✓	**97.57**	**97.48**	**97.56**	**97.36**	**97.47**	**97.36**	**97.50**	97.17	**97.43**
step200	✗	97.31	97.31	97.31	97.31	97.31	**97.31**	97.31	**97.31**	97.31
	✓	**97.47**	**97.39**	**97.43**	**97.33**	**97.42**	97.21	**97.36**	97.19	**97.35**

Table A53. **Comparison (↑) on online DAGM** in the same format as Tab. A51.

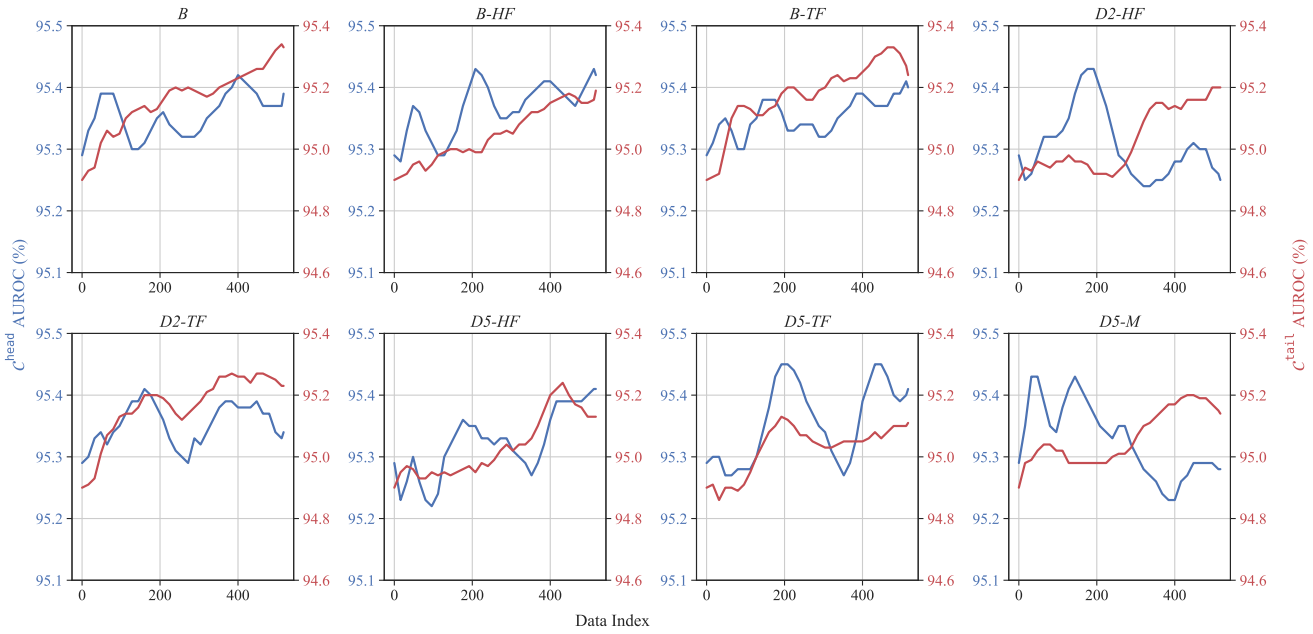


Figure A5. **Performance curves** of LTOAD in pixel-level AUROC on $\mathcal{C}^{\texttt{head}}$ and $\mathcal{C}^{\texttt{tail}}$. They are offline-trained on MVTec *exp100* [34] and online-trained on our online benchmarks.

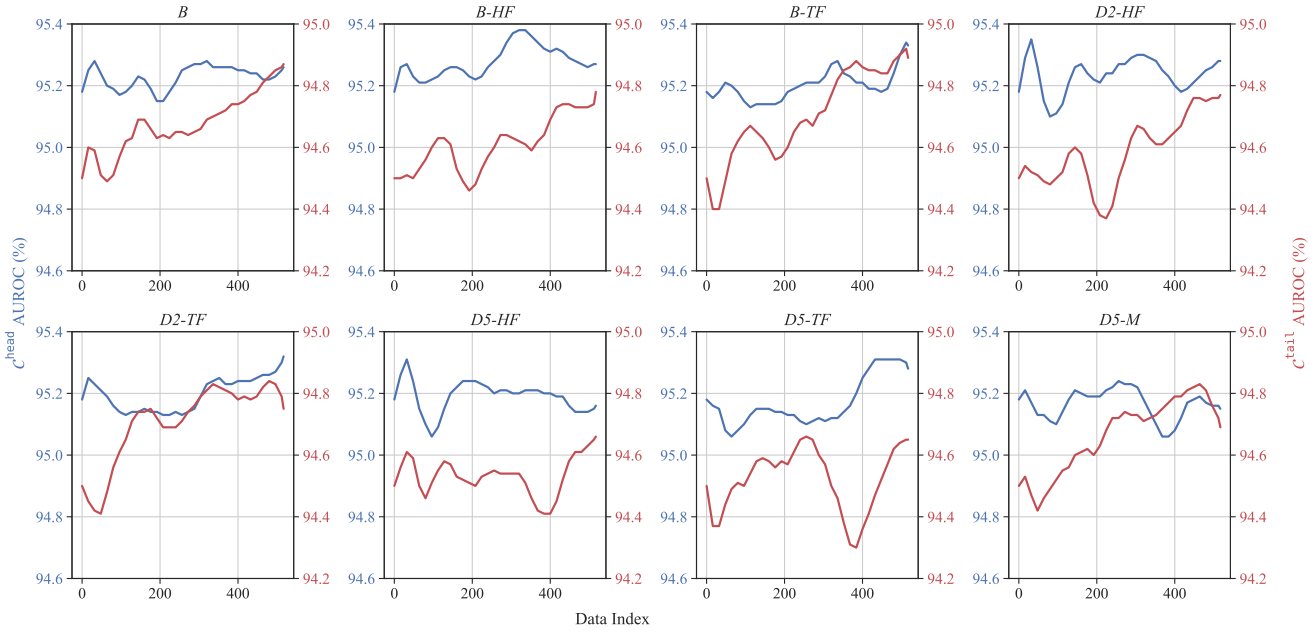


Figure A6. **Performance curves** in the same format as Tab. A51. They are offline-trained on MVTec *exp200* [34].

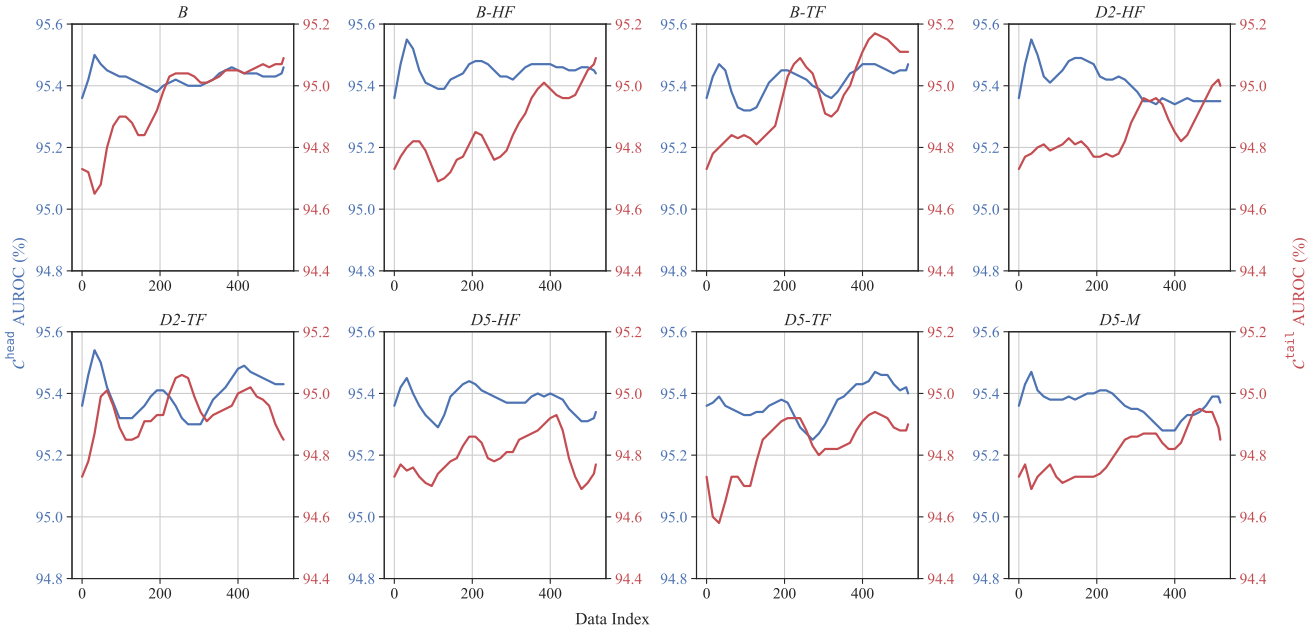


Figure A7. **Performance curves** in the same format as Tab. A51. They are offline-trained on MVTec *step100* [34].

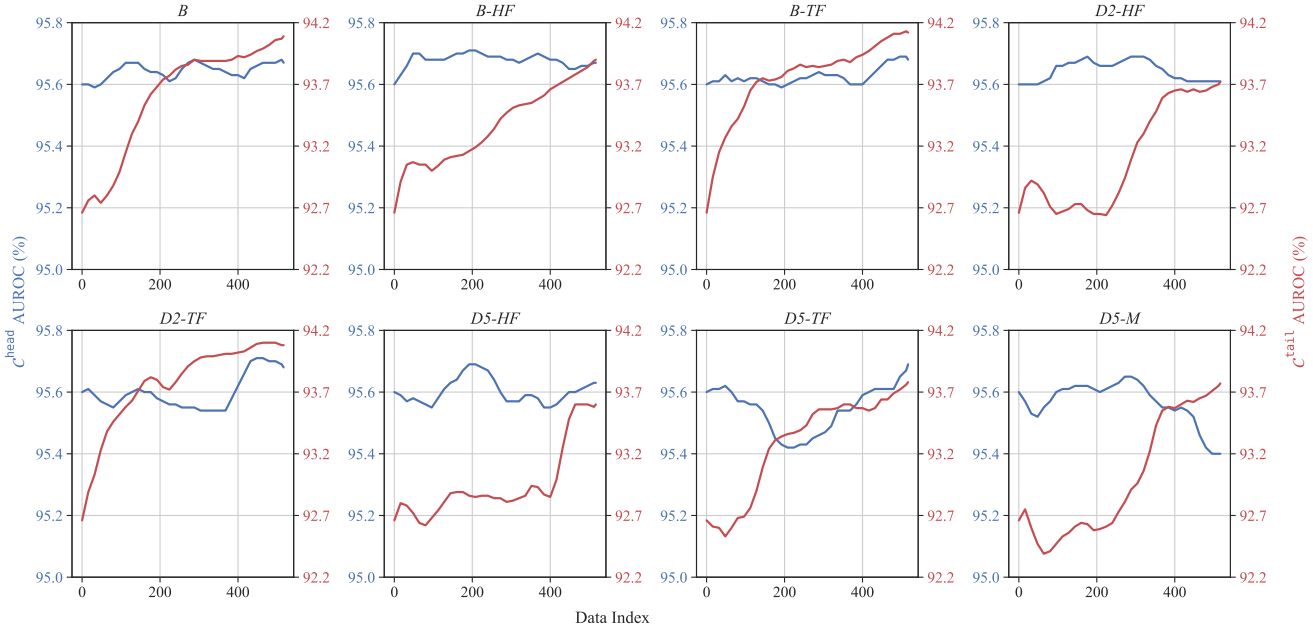


Figure A8. **Performance curves** in the same format as Tab. A51. They are offline-trained on MVTec *step200* [34].

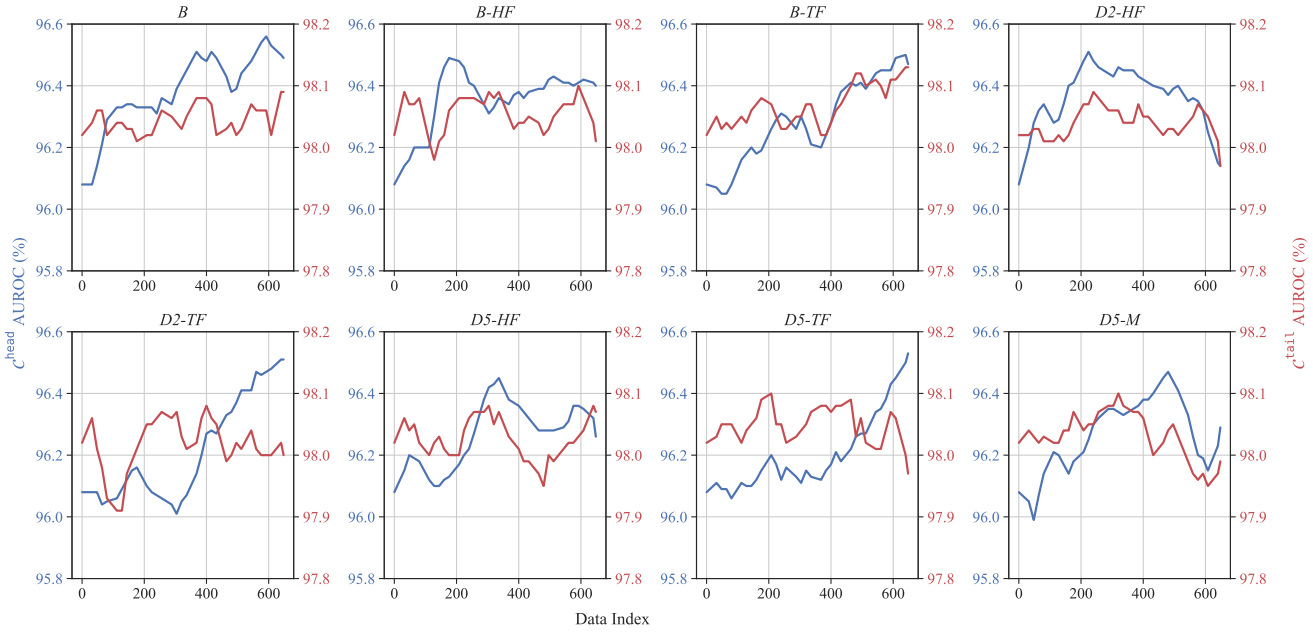


Figure A9. **Performance curves** in the same format as Tab. A51. They are offline-trained on VisA *exp100* [34].

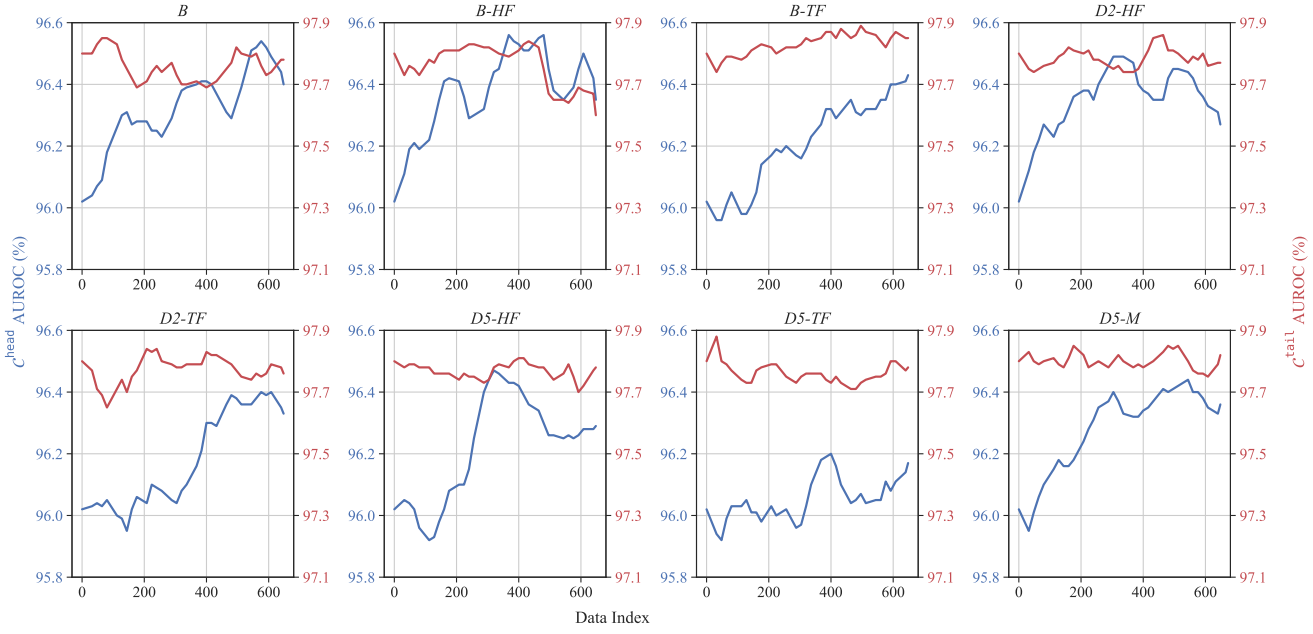


Figure A10. **Performance curves** in the same format as Tab. A51. They are offline-trained on VisA *exp200* [34].

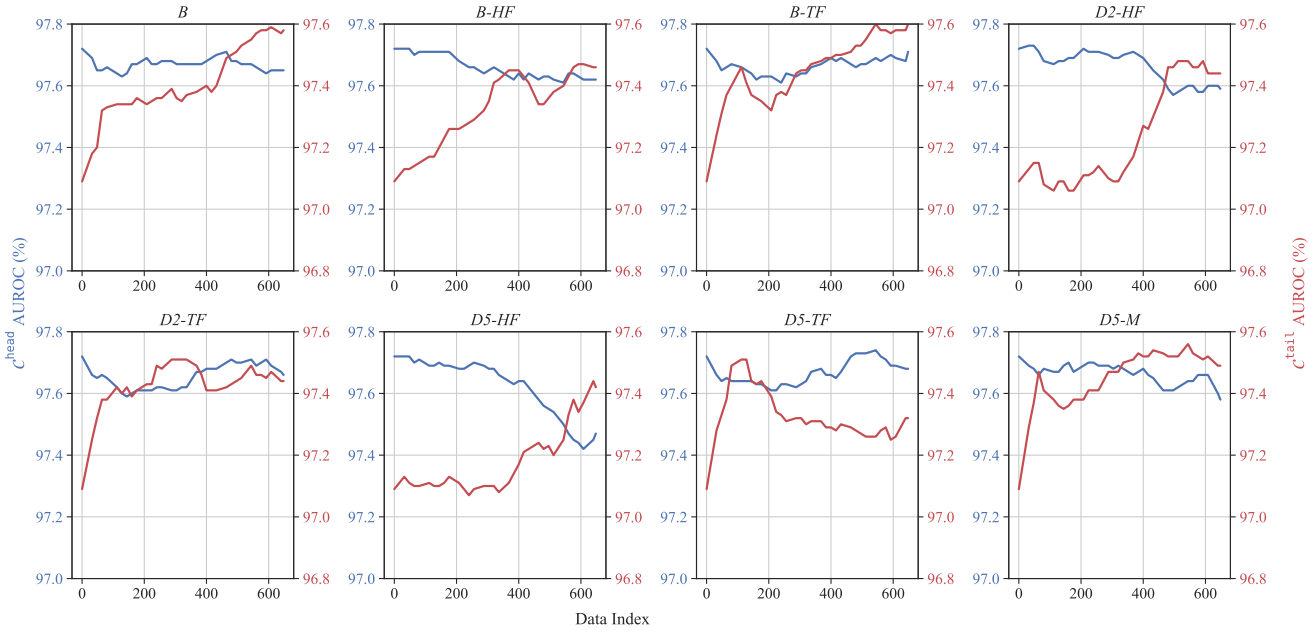


Figure A11. **Performance curves** in the same format as Tab. A51. They are offline-trained on VisA *step100* [34].

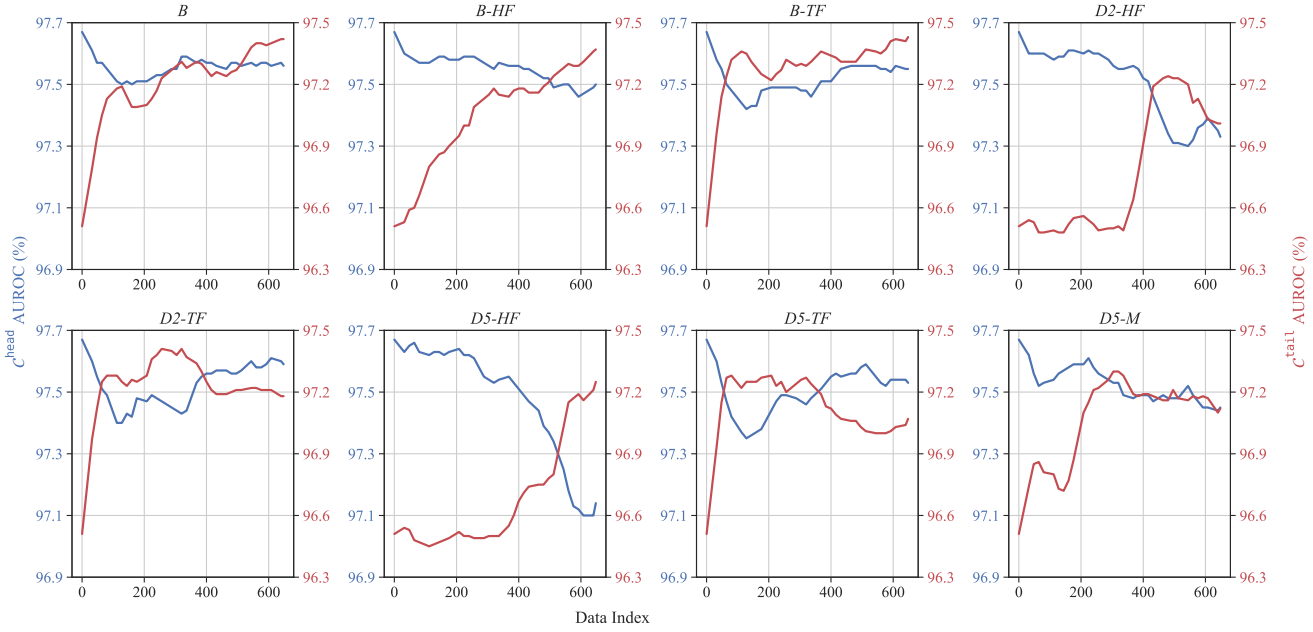


Figure A12. **Performance curves** in the same format as Tab. A51. They are offline-trained on VisA *step200* [34].

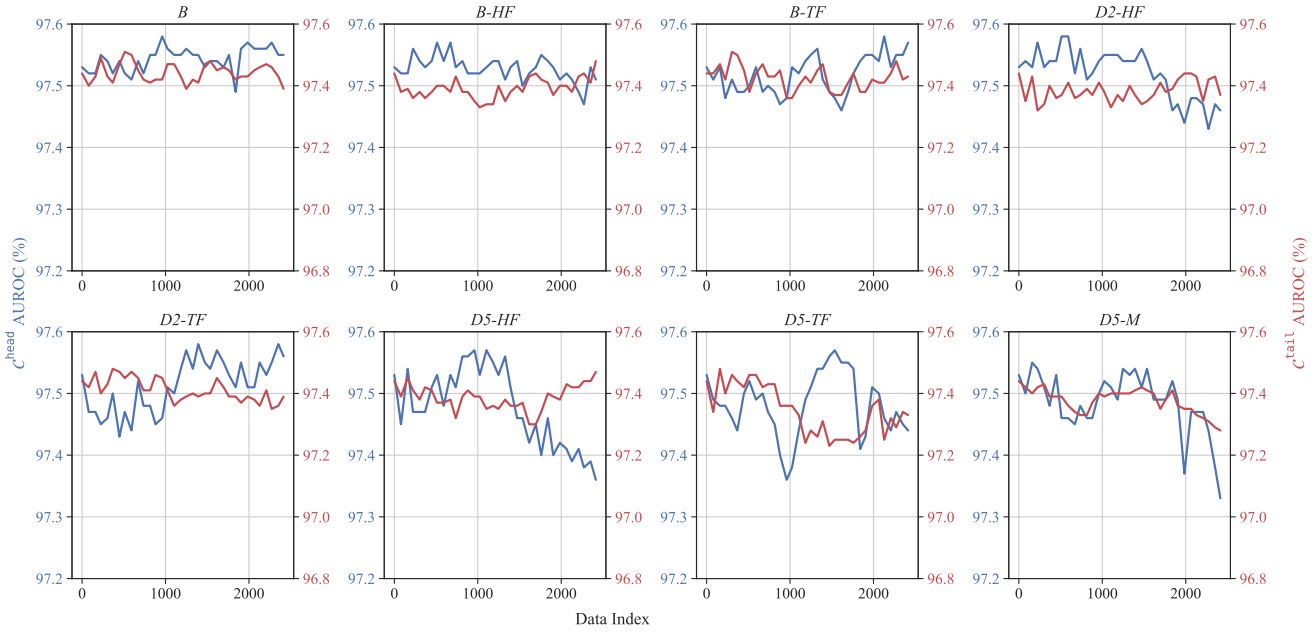


Figure A13. **Performance curves** in the same format as Tab. A51. They are offline-trained on DAGM *exp100* [34].

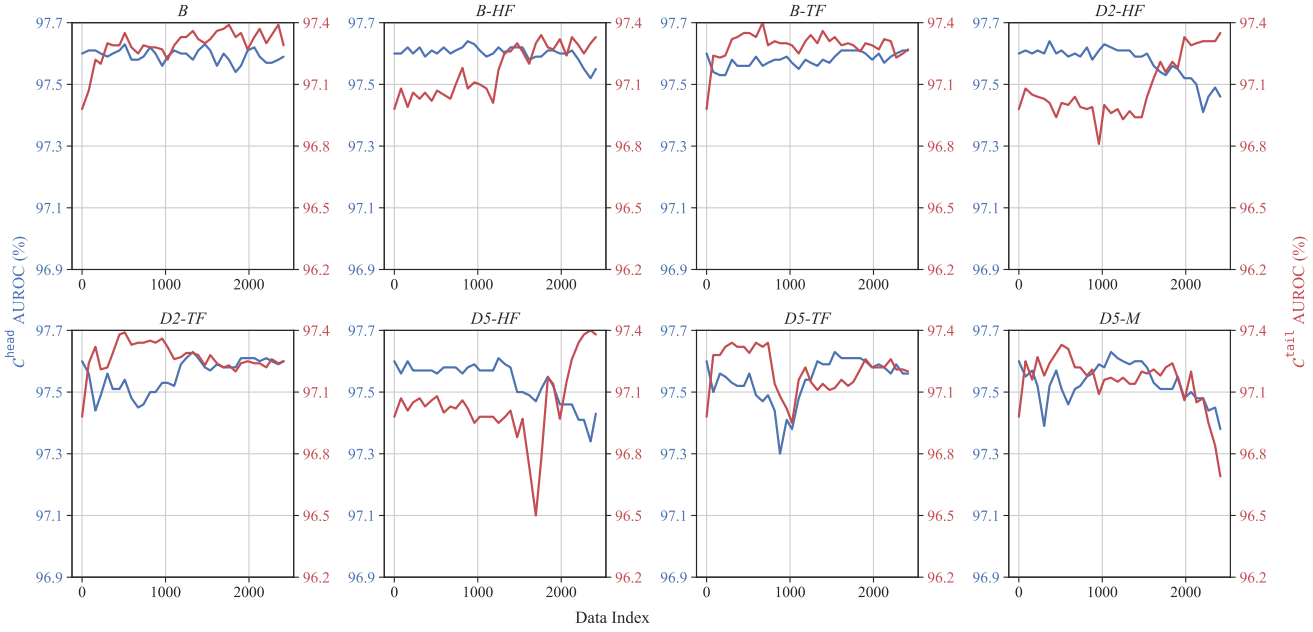


Figure A14. **Performance curves** in the same format as Tab. A51. They are offline-trained on DAGM *exp200* [34].

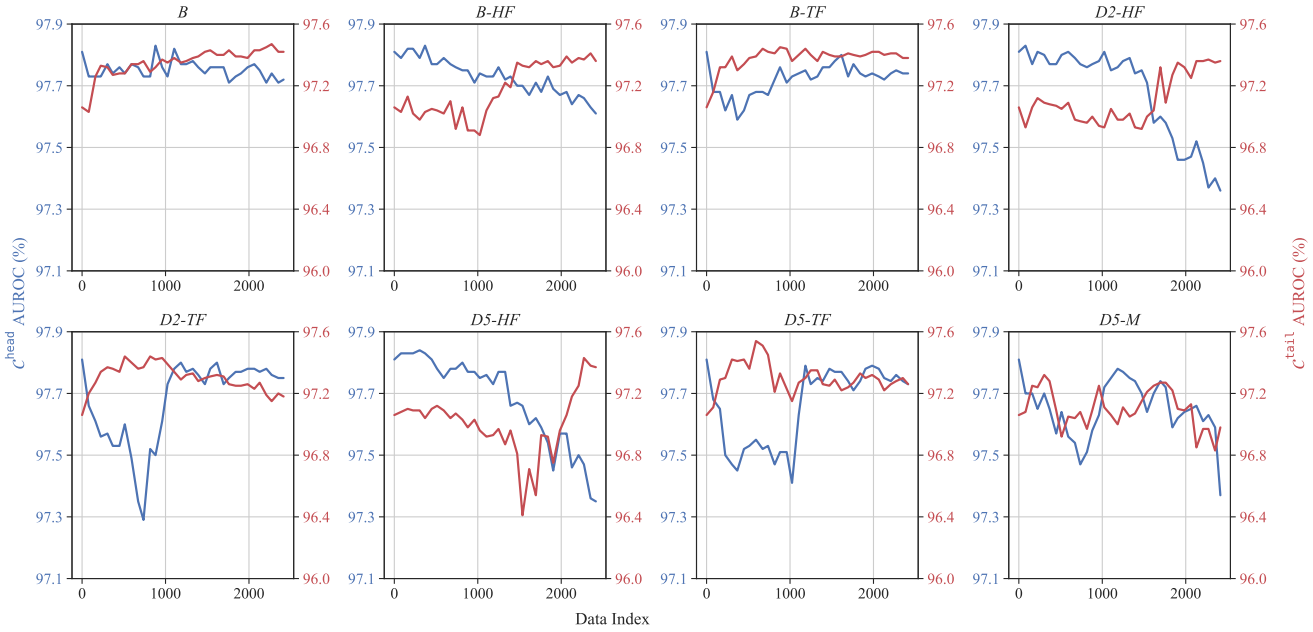


Figure A15. **Performance curves** in the same format as Tab. A51. They are offline-trained on DAGM *step100* [34].

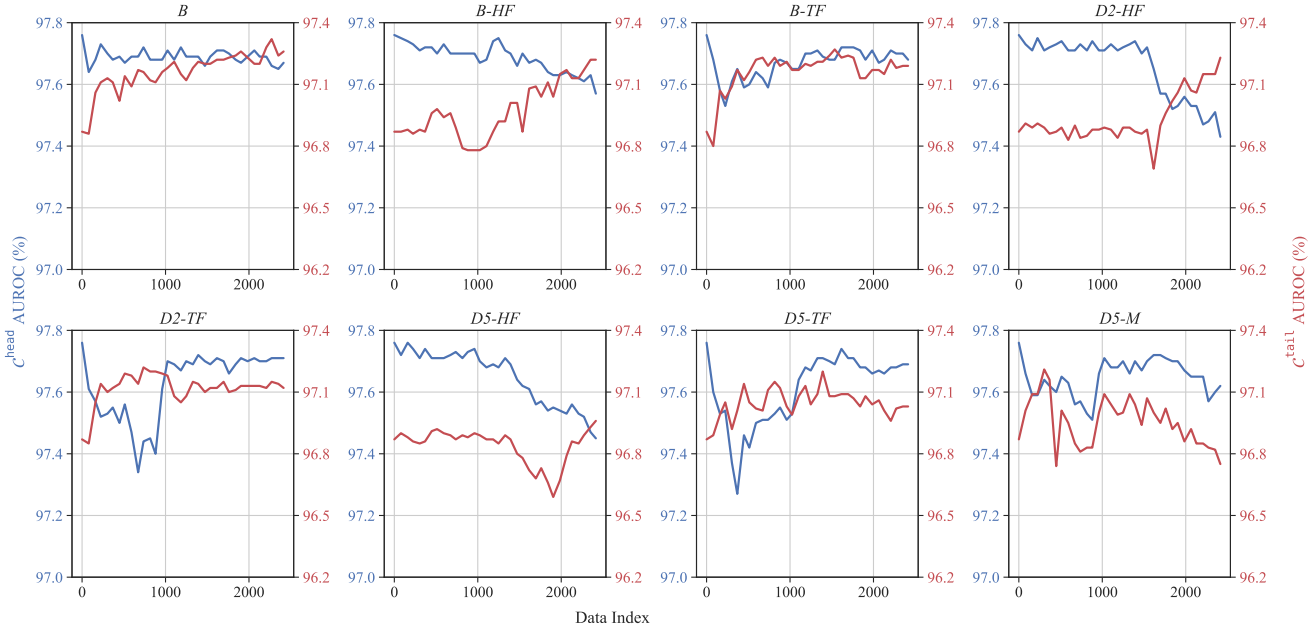


Figure A16. **Performance curves** in the same format as Tab. A51. They are offline-trained on DAGM *step200* [34].