RECONSTRUCTION-BASED ANOMALY DETECTION IN POSITIVE-UNLABELED REVERSE DISTILLATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Reverse distillation is one of the strongest paradigms for visual anomaly detection, training a student decoder to reconstruct a frozen teacher encoder's representations so that unseen anomalies yield disproportionately large reconstruction errors. However, this paradigm implicitly assumes a training pool of only normal samples. In practice, such pool is often contaminated, modeled as a mixture of normal and anomalous distributions at an unknown rate, and training directly on this mixture provides no pressure to separate the two populations. Positive-unlabeled learning (PUL) offers a natural way to address this, incorporating a small labeled anomalous set to guide this separation. However, existing PUL approaches largely rely on classification-level objectives that offer little sense of where an anomaly is located. Instead, we build directly on the spatial structure of reconstruction error in feature space, which already carries information about where an anomaly might be located, without requiring pixel-level annotation. We propose an objective that thresholds each image's own feature-space reconstruction error into anomalous and normal regions and applies separation pressure between them. We apply this pressure asymmetrically. For labeled anomalies, whose status is certain, we push the two regions apart in both directions. For unlabeled samples, we only push the flagged region's error down. Evaluated on Real-IAD, an industrial anomaly benchmark, our method consistently improves over the uncorrected baseline and recovers much of the performance gap to an oracle trained without contamination.

1 INTRODUCTION

Reconstruction-based anomaly detection trains a model to reconstruct normal inputs or normal feature representations, using reconstruction quality itself as the anomaly signal at test time. Reverse distillation (Deng & Li, 2022) instantiates this with a frozen, pretrained teacher encoder and a trainable student that reconstructs the teacher's features from a bottlenecked summary. Combined with strong self-supervised backbones such as DINOv2 (Oquab et al., 2024), recent instantiations of this paradigm (Guo et al., 2025) are among the strongest current approaches to industrial anomaly detection, producing not just an image-level score but a full spatial error map that localizes the anomaly.

However, this paradigm assumes every training sample is normal, an assumption that rarely holds in the real world. Indeed, verifying that a large training set is entirely defect-free is itself a labeling problem. Manual labeling is often expensive, especially in dynamic industrial settings with varied product lines, where constructing large datasets of non-defective objects can be challenging. Thus, unlabeled training sets are typically *contaminated* with a minority fraction of hidden anomalies. In training on this sort of mixture datasets, no signal to separate the two populations is given, and the model is left learning patterns of hidden anomalies, along with the normal majority.

Positive-unlabeled learning (PUL, Elkan & Noto, 2008) offers a natural correction to this unwanted behavior: alongside the contaminated pool, we assume access to a small labeled set of confirmed anomalies (often possible to obtain in industrial settings from past quality-control logs). The anomaly set is then used as a correction term to "de-contaminate" the unlabeled pool, filtering the effect of anomalous patterns during learning. Anomaly detection with contaminated training data has been addressed through latent outlier inference (Qiu et al., 2022), active labeling (Li et al., 2023), memory-bank denoising (Jiang et al., 2022), and, closest to our setting, positive-unlabeled correction (Takahashi et al., 2026). In these methods, the correction acts on a single anomaly score

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

per image. Even where the detector's reconstruction error is computed per-pixel internally, as in an autoencoder (Takahashi et al., 2026), the correction itself acts only after this is collapsed into one number, comparing whole images against one another rather than distinguishing between regions within the same image. This discards exactly the spatial structure reconstruction-based detectors already provide for free: an error map that is dense over the image. Furthermore, this correction fundamentally relies on knowing a contamination rate *i.e.*, the ratio of anomalous images in the unlabeled pool, a rather unrealistic assumption in real-world settings.

In this work, we combine the strengths of both paradigms above: the spatial error map reconstruction-based detectors already provide, and PU learning's ability to correct for contamination. We build a correction for the error maps generated by reconstruction-based detectors, and we do so directly in the feature space, rather than in raw pixel space. This choice is not incidental: reconstructing raw pixels forces a model to spend capacity on high-frequency detail, texture, and lighting that carries little information about whether a region is defective, and lets a model reach partial success by reproducing low-level statistics without ever representing the content that actually separates normal from anomalous. Because the encoder was already trained to be invariant to nuisance factors like lighting and texture noise, reconstructing its features means the detector only has to worry about the variation that matters. The same argument motivates recent self-supervised world models such as I-JEPA and V-JEPA (Assran et al., 2023; Bardes et al., 2024), which predict masked regions in representation space rather than pixel space to avoid wasting capacity on unpredictable low-level detail.

Notably, we operate on this feature-space error map directly, at the pixel level. For each image, we threshold its own reconstruction error into a flagged region and its complement. We then apply separation pressure between the two regions, rather than collapsing the *whole* error map into one scalar. Additionally, standard reconstruction-based detectors reduce this map with a fixed, image-agnostic rule, *e.g.*, a flat average over the whole map, or a fixed top-k cutoff, applied identically to every image regardless of its own error distribution. Instead, we use Otsu's method (Otsu, 1979), an equally fast procedure that recomputes the split from each image's own distribution. We apply this separation pressure asymmetrically. For labeled anomalies, whose status is certain, we push the flagged region's error up and the rest down. For unlabeled samples, the same split instead isolates only the hardest-to-reconstruct region, not a confirmed defect, and we only ever push it down, consistent with ordinary hard-region reconstruction training rather than any claim about its true status.

We evaluate our method on the Real-IAD (Wang et al., 2024) dataset, a large-scale industrial benchmark, using a Dinomaly-style backbone. Unlike the simplified benchmarks typically used to evaluate PU-based anomaly detectors, Real-IAD reflects the scale and visual complexity of real industrial inspection. Since standard anomaly benchmarks are built for the one-class setting, we construct our own PU split with a naturally occurring, category-dependent contamination ratio. Our approach consistently improves image-level AUROC over the uncorrected baseline and recovers much of the gap to an oracle trained on decontaminated data, without access to the contamination labels the oracle uses.

In short, our contributions are: (i) a PU objective for reconstruction-based anomaly detectors that applies the correction to regions of each image's error map, selected by a per-image threshold, rather than to a single whole-image score, without pixel-level annotation; (ii) an asymmetric design that creates within-image contrast on labeled anomalies through a bounded per-sample hinge and an anchor term, while applying only one-sided pressure to the unlabeled pool; (iii) an objective that does not require the contamination rate, with a single hyperparameter setting across categories in our experiments.

2 Related work

2.1 Positive-Unlabeled Learning

Positive-unlabeled learning addresses the setting where only a subset of the positive class is labeled and all remaining data, positive or negative, is unlabeled. The foundational formulation is due to Elkan & Noto (2008), who showed that a classifier trained to distinguish labeled positives from unlabeled data can be corrected into an estimate of the true positive-vs-negative classifier under

a missing-completely-at-random labeling assumption. A central theoretical thread in this line of work is unbiased risk estimation. Since the unlabeled pool is a known mixture of positives and negatives, an unbiased estimator of the classification risk can be recovered algebraically from the positive and unlabeled empirical risks alone, without ever observing a labeled negative. du Plessis et al. (2015) first derived this estimator, and Kiryo et al. (2017) identified that it can go negative for flexible models such as deep networks, causing severe overfitting, and introduced the non-negative correction that clips the estimator at zero.

A second thread moves PU correction from a risk estimator on a fixed classifier into the representation itself. Chen et al. (2020) combine self-paced sample selection with a teacher-student consistency term to stabilize deep PU training. Zhao et al. (2022) reformulate PU classification as matching a predicted label distribution against an estimated target distribution rather than correcting a pointwise risk. Acharya et al. (2022) and Yuan et al. (2025) bring PU into the contrastive learning setting, the latter weighting and mining hard negatives specifically among unlabeled examples.

What unites both paragraphs above, classical or deep, is that the object being labeled is treated as atomic: every method outputs one class decision per example, with no notion of *where within* that example the positive evidence sits.

2.2 Spatial Anomaly Detection

Several paradigms produce spatial anomaly evidence as a byproduct of their detection mechanism, evaluated pixel-by-pixel rather than pooled into a single decision at training time. Roth et al. (2022) introduce PatchCore, a memory-bank method that scores each patch by its distance to a bank of normal embeddings. Liu et al. (2023) instead inject noise into the embedding space and train a discriminator to separate normal from perturbed-normal features, a hybrid discriminative approach that likewise localizes anomalies at the patch level without pixel supervision.

We build on the reconstruction paradigm: a model learns to reconstruct normal inputs or features, and reconstruction quality itself becomes the anomaly signal. Knowledge-distillation variants use a frozen, pretrained teacher and a trainable student asked to match the teacher's representations only on normal data, so the mismatch at test time localizes anomalies directly. Deng & Li (2022) introduce reverse distillation, reversing the usual student-teacher direction so the student decodes the teacher's encoded representation rather than imitating its forward pass. Tien et al. (2023) strengthen this with a multi-scale fusion bottleneck and improved training recipe. Guo et al. (2025) instantiate the same principle on a frozen DINOv2 Oquab et al. (2024) backbone with a noised bottleneck and linear-attention decoder as a single multi-class model. We adopt the reverse-distillation formulation and the Dinomaly backbone directly, motivated by its state-of-the-art accuracy on strong self-supervised backbones and by its error map, a per-location teacher-student disagreement, composing naturally with a per-image adaptive threshold.

2.3 Contamination-aware Anomaly Detection

A separate line of work addresses the same contamination problem within anomaly detection specifically, rather than as general classification. Semi-supervised detectors such as Deep SAD (Ruff et al., 2020) extend one-class deep SVDD (Ruff et al., 2018) with a small labeled anomaly pool, pulling normal embeddings toward a center and pushing labeled anomalies away from it, but implicitly assume the unlabeled pool used for the one-class term is itself clean. Fewer works instead treat the unlabeled pool itself as contaminated. Qiu et al. (2022) introduce Latent Outlier Exposure (LOE), treating each unlabeled point's label as a latent variable and alternating between inferring it and optimizing the base detector, without any labeled anomaly pool at all. Li et al. (2023) build on LOE with a query strategy that decides which unlabeled examples are most worth labeling under a limited budget, combining LOE's contamination-awareness with a small labeled pool. Both produce a single scalar score per image, with no spatial error map to localize against. Jiang et al. (2022) address noisy unlabeled data in the memory-bank paradigm by filtering unreliable patch embeddings before they enter the memory bank, denoising the pool rather than correcting a training objective. Takahashi et al. (2026) integrate positive-unlabeled correction with deep anomaly detectors such as autoencoders, approximating the anomaly score for normal data jointly from the unlabeled and labeled-anomaly pools rather than assuming the unlabeled pool is already normal. However, their

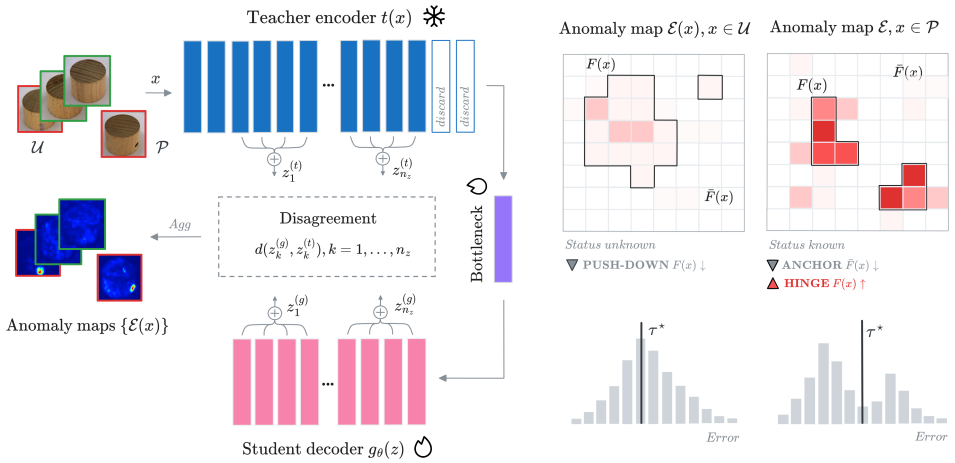


Figure 1: Overview of the method. Left: frozen DINOv2 teacher and trainable student produce a spatial error map. Right: for a labeled anomaly $\boldsymbol{x} \in \mathcal{P}$, Otsu thresholding isolates $F(\boldsymbol{x})$ correlating with the true defect; hinge pushes its error up, anchor pushes $\bar{F}(\boldsymbol{x})$ down. For $\boldsymbol{x} \in \mathcal{U}$, the same split instead isolates the hardest-to-reconstruct region rather than a confirmed defect, and this region is only ever pushed down, exactly as ordinary hard-region mining already does, since we cannot be sure it is anomalous.

correction operates on raw pixels rather than a pretrained encoder's features, and is computed on a single whole-image score with no mechanism to separate defective from non-defective regions.

3 Methodology

The objective of our work is to train a detector that takes as input an image sample $x \in \mathbb{R}^{C \times H \times W}$, and returns a per-pixel anomaly score map $\mathcal{E} \in \mathbb{R}^{H \times W}_{\geq 0}$. In particular, we are given two training sets: an *unlabeled set* $\mathcal{U} = \{\boldsymbol{x}_i\}_{i=1}^{N_u}$ containing both normal and anomalous samples, and a (smaller) *positive set* $\mathcal{P} = \{\boldsymbol{x}_j\}_{j=1}^{N_p}$ containing only confirmed anomalies. In training our model, therefore, we want to learn a condition of normality from $\mathcal{U}$, exploiting $\mathcal{P}$ to filter out the effect of anomalous samples.

In the following, we discuss our proposed approach for anomaly detection. We start by describing the detection architecture, and proceed to illustrate its per-pixel positive-unlabeled training procedure.

3.1 Inference: Reverse Distillation

For the detector, we adopt a reverse distillation (RD) teacher-student architecture (*cf.* Fig. 1). In ordinary distillation, a model (the *student*) is trained to imitate a frozen pre-trained model (the *teacher*) on a shared forward pass. In RD, instead, the teacher is an encoder $t(\boldsymbol{x})$ outputting image embeddings $\boldsymbol{z}$, and the student is a decoder $g_\theta(\boldsymbol{z})$ tasked to perform reconstruction from the embedding, thus acting as the functional inverse of the teacher. To prevent the student from trivially copying $\boldsymbol{z}$ rather than reconstructing it, a bottleneck sits between encoder and decoder, compressing $\boldsymbol{z}$ and injecting noise before the student ever sees it, so that g_θ must recover the teacher's representation from this degraded summary rather than from $\boldsymbol{z}$ itself. In classic AD applications, it only ever learns to do so successfully on normal data. An anomalous input therefore produces a teacher representation the student has never had to invert, and reconstruction degrades accordingly.

As in modern RD approaches, we work with transformer architectures so that every representation lies in a common identically-sized space $\mathbb{R}^{C_z \times H_z \times W_z}$. The reconstruction error is generated by computing, first, the disagreement of each pair $(\boldsymbol{z}_k^{(t)}, \boldsymbol{z}_k^{(g)})$, $k = 1, \ldots, n_z$ at every spatial location

(i, j), via cosine distance between their channel-wise feature vectors:

$$d_{kij} = 1 - \frac{\langle \boldsymbol{z}_{kij}^{(t)}, \boldsymbol{z}_{kij}^{(g)} \rangle}{\|\boldsymbol{z}_{kij}^{(t)}\| \|\boldsymbol{z}_{kij}^{(g)}\|} \in [0, 2], \quad i = 1, \ldots, H_z,\ j = 1, \ldots, W_z. \tag{1}$$

The distances so computed are then averaged across all embeddings to obtain the anomaly map $\mathcal{E}(\boldsymbol{x}) = (\frac{1}{n_z} \sum_{k=1} d_{kij})_{ij}$. $\mathcal{E}(\boldsymbol{x})_{ij} \in [0, 2]$ is our per-pixel evidence that location (i, j) of image $\boldsymbol{x}$ is anomalous. In a detector properly trained to learn normality, the reconstruction of the anomalous region is missed, thus generating a large error in the corresponding locations of the anomaly map.

3.2 Training: Pixel-level Positive-Unlabeled Learning

Classical Setting. To train our anomaly detector, we need to filter out the effect of anomalous samples in our unlabeled set. To do so, we adopt a positive-unlabeled learning approach. Let $P_n(\boldsymbol{x})$ and $P_a(\boldsymbol{x})$ denote the true, unknown distributions of normal and anomalous images, respectively. In PUL, a sample $\boldsymbol{x} \in \mathcal{U}$ is assumed to be drawn from a mixture of the two:

$$P_u(\boldsymbol{x}) = (1 - \alpha) P_n(\boldsymbol{x}) + \alpha P_a(\boldsymbol{x}), \tag{2}$$

where $\alpha \in [0, 1)$ denotes the (unknown) *contamination ratio*, *i.e.*, the fraction of observations in $\mathcal{U}$ that is secretly anomalous. In practice, neither P_n nor P_a is directly observed. The classical PUL task, then, consists of using $\mathcal{P}$ as a proxy to estimate P_a, retrieving $P_n(\boldsymbol{x}) = (1 - \alpha)^{-1}(P_u(\boldsymbol{x}) - \alpha P_a(\boldsymbol{x}))$.

Anomaly Detection Setting. In this work, we go beyond the mere identification of anomalous samples, and we focus on the localization of the anomaly within an image. To this aim, we adapt the PUL setup and develop a pixel-level PUL anomaly detection framework, as follows. Starting from the anomaly map $\mathcal{E}(\boldsymbol{x})$, we threshold it and extract a *flagged region* $F(\boldsymbol{x}) = \{(i, j) \mid \mathcal{E}(\boldsymbol{x})_{ij} \geq \tau(\boldsymbol{x})\}$, *i.e.*, a set of "good" candidate locations for potential anomalies $\boldsymbol{x}$. For the threshold τ, a fixed top-$k\%$ split is the simplest option, used in recent works such as Guo et al. (2025), but we find it too simplistic, as it does not scale with the anomaly size. Instead, we recompute the split from each image's own error distribution, using Otsu's method (Otsu, 1979), a classical, parameter-free procedure for exactly this purpose.

After extracting the flagged region, we average its values and obtain a *local anomaly score* $R^+(\boldsymbol{x}) = \frac{1}{|F(\boldsymbol{x})|} \sum_{(i,j) \in F(\boldsymbol{x})} \mathcal{E}(\boldsymbol{x})_{ij}$, giving a rough estimate of the presence of an anomaly in $F(\boldsymbol{x})$. From Eq. (2), it follows that (*cf.* §A for a proof):

$$\mathbb{E}_{P_n}[R^+(\boldsymbol{x})] = (1 - \alpha)^{-1}(\mathbb{E}_{P_u}[R^+(\boldsymbol{x})] - \alpha \mathbb{E}_{P_a}[R^+(\boldsymbol{x})]). \tag{3}$$

Similarly to the general case, this expression constitutes a natural loss term for our objective, estimating $\mathbb{E}_{P_u}$ and $\mathbb{E}_{P_a}$ with $\mathcal{U}$ and $\mathcal{P}$ respectively. Note, however, that the naive empirical minimization of such an objective does not provide any guidance to the scores of the anomaly map excluded from the flagged region. This constitutes a fundamental drawback since the model is not restricted from pushing the scores in an anomaly map *globally* to zero in the unlabeled set, and to the maximum score value (2, in our case) for the anomalous samples.

To prevent such behavior, we adapt Eq. (3) accordingly to obtain the following loss used in training our models:

$$\begin{aligned} \mathcal{L}(\boldsymbol{x}; \theta) &= \mathcal{L}_{\mathcal{U}} + \beta \cdot \mathcal{L}_{\mathcal{P},\text{hinge}} + \gamma \cdot \mathcal{L}_{\mathcal{P},\text{anchor}} \\ &= \mathbb{E}_{\boldsymbol{x} \in \mathcal{U}}[R^+(\boldsymbol{x})] + \beta \cdot \mathbb{E}_{\boldsymbol{x} \in \mathcal{P}}[\max(0, m - R^+(\boldsymbol{x}))] + \gamma \cdot \mathbb{E}_{\boldsymbol{x} \in \mathcal{P}}[R^-(\boldsymbol{x})], \end{aligned} \tag{4}$$

where $\beta > 0$, $\gamma > 0$ and $m \in [0, 2]$ are hyperparameters, and R^- is the local anomaly score computed on the *unflagged region*, *i.e.*, the complement of $F(\boldsymbol{x})$. The anchor term encourages keeping low scores on the unflagged region in the positive samples. The hinge term acts as a regularizer: it pushes the flagged region's score toward the margin, but stops contributing once that margin is reached, so already-separated anomalies stop competing for gradient with harder ones still below it. Note that when the margin is not active, the two first terms differ from the empirical error related to Eq. (3) only by a constant bias βm and a factor $(1 - \alpha)^{-1}$. Additionally, in the experimental section we compare this form of the hinge term against an alternative in which the margin is applied to the batch mean rather than to each sample individually.

Table 1: **Main quantitative results.** Test-set Image-AUROC, Pixel-AUROC, and PRO across Real-IAD categories. RD treats the unlabeled pool as clean; SoftPatch uses a memory bank with soft-weighted patches; Oracle removes contamination from $\mathcal{U}$ using ground-truth labels and serves as a diagnostic clean-data reference. Best score across non-oracle models is in bold.

		RD			SoftPatch			Ours			Oracle		
Category	α	AUC-I	AUC-P	PRO	AUC-I	AUC-P	PRO	AUC-I	AUC-P	PRO	AUC-I	AUC-P	PRO
audiojack	0.212	0.863	0.995	0.951	0.856	0.983	0.896	**0.920**	**0.997**	**0.959**	0.934	0.997	0.957
bottle_cap	0.279	0.854	0.994	0.960	0.955	0.997	**0.980**	**0.958**	**0.999**	**0.980**	0.983	0.999	0.986
button_battery	0.365	0.858	**0.995**	0.941	0.833	0.994	0.919	**0.933**	**0.995**	**0.960**	0.914	0.996	0.941
end_cap	0.372	0.735	0.990	0.952	0.803	0.971	0.890	**0.908**	**0.996**	**0.967**	0.895	0.996	0.970
eraser	0.248	0.902	0.997	0.950	**0.941**	0.995	0.958	0.939	**0.998**	**0.972**	0.949	0.998	0.959
fire_hood	0.226	0.824	0.996	0.902	0.824	0.989	0.838	**0.911**	**0.998**	**0.946**	0.899	0.998	0.933
mint	0.353	0.721	0.975	0.800	0.761	**0.989**	0.806	**0.803**	**0.989**	**0.880**	0.817	0.989	0.844
mounts	0.223	0.898	0.995	0.928	0.932	0.985	0.890	**0.940**	**0.998**	**0.966**	0.968	0.998	0.948
pcb	0.389	0.893	**0.998**	0.977	0.895	0.996	0.957	**0.907**	**0.998**	**0.984**	0.940	0.999	0.977
phone_battery	0.266	0.904	**0.999**	0.971	0.899	0.993	0.952	**0.936**	**0.999**	**0.979**	0.954	0.999	0.973
plastic_nut	0.206	0.907	0.998	0.976	0.917	0.995	0.965	**0.940**	**0.999**	**0.984**	0.955	0.999	0.975
plastic_plug	0.241	0.924	0.996	0.941	0.931	0.995	0.945	**0.944**	**0.998**	**0.969**	0.963	0.997	0.954
porcelain_doll	0.220	0.871	0.992	0.937	0.876	0.989	0.931	**0.915**	**0.995**	**0.954**	0.926	0.994	0.942
regulator	0.177	0.864	**0.999**	**0.985**	0.879	0.997	0.964	**0.897**	0.998	0.979	0.922	0.999	0.984
rolled_strip_base	0.427	0.933	**0.999**	0.982	0.949	0.993	0.975	**0.981**	**0.999**	**0.992**	0.998	0.998	0.982
sim_card_set	0.350	0.933	0.997	0.947	0.949	0.997	0.958	**0.958**	**0.998**	**0.963**	0.979	0.997	0.929
switch	0.336	0.950	**0.976**	0.949	0.956	0.887	0.914	**0.978**	0.970	**0.952**	0.993	0.996	0.970
tape	0.242	0.945	0.998	0.972	0.952	0.996	0.964	**0.974**	**0.999**	**0.977**	0.989	0.999	0.979
terminalblock	0.339	0.963	**0.999**	0.986	0.981	0.998	0.987	**0.983**	**0.999**	**0.991**	0.998	0.999	0.991
toothbrush	0.345	0.839	0.964	0.816	**0.884**	**0.981**	**0.913**	**0.884**	0.979	0.897	0.919	0.980	0.877
toy	0.369	0.752	0.963	0.886	0.864	**0.972**	0.849	**0.893**	0.963	**0.911**	0.894	0.977	0.902
toy_brick	0.264	0.718	0.974	0.759	**0.847**	0.986	**0.865**	0.782	**0.987**	0.837	0.866	0.990	0.873
transistor1	0.359	0.853	0.989	0.950	0.889	0.988	0.947	**0.964**	**0.993**	**0.975**	0.979	0.994	0.969
u_block	0.200	0.934	0.998	0.954	0.955	0.997	0.963	**0.957**	**0.999**	**0.970**	0.971	0.999	0.949
usb	0.301	0.902	0.998	0.975	0.931	0.995	0.963	**0.953**	**0.999**	**0.982**	0.967	0.999	0.982
usb_adaptor	0.255	0.834	0.997	0.944	0.817	0.991	0.929	**0.898**	**0.998**	**0.951**	0.914	0.998	0.944
vcpill	0.246	0.872	0.989	0.902	0.887	**0.991**	0.906	**0.927**	**0.991**	**0.930**	0.938	0.994	0.928
wooden_beads	0.293	0.844	0.993	0.902	0.895	0.983	0.871	**0.944**	**0.996**	**0.941**	0.936	0.996	0.925
woodstick	0.180	0.814	0.989	0.912	0.715	0.935	0.679	**0.910**	**0.995**	**0.953**	0.895	0.994	0.945
zipper	0.392	0.978	0.985	0.946	0.976	0.983	0.955	**0.988**	**0.991**	**0.966**	0.999	0.996	0.976
Mean		0.869	0.991	0.932	0.892	0.985	0.918	**0.928**	**0.994**	**0.956**	0.942	0.996	0.949

One might be tempted to add an additional term $\mathbb{E}_{\boldsymbol{x}\in\mathcal{U}}[R^{-}(\boldsymbol{x})]$ penalizing high anomaly scores for unflagged regions in the unlabeled pool. However, pushing both regions down – mirroring the anchor symmetrically – is arithmetically close to minimizing the mean of the whole image, discarding the reason a region-selective objective was introduced at all.

4 Experiments

In this section, we describe the experimental setup used to assess the performance of our method and discuss the results we have obtained. Additionally, we conduct several ablation studies to gain additional insight into the motivation behind the reported performance.

4.1 Experimental Settings

We evaluate our method on Real-IAD (Wang et al., 2024), a *large-scale* industrial anomaly detection dataset spanning 30 object categories with diverse defect types, each with pixel-level annotations. In particular, we adapt it to the PUL setting so that we isolate a small subset of anomalous images to simulate training and validation sets $\mathcal{P}_{\text{train}}$, $\mathcal{P}_{\text{val}}$, and we use the rest to build $\mathcal{U}_{\text{train}}$, $\mathcal{U}_{\text{val}}$, $\mathcal{U}_{\text{test}}$. Following Dinomaly (Guo et al., 2025), we adopt a DINOv2 teacher encoder (Oquab et al., 2024), and the student is a compact trainable network that reconstructs the teacher's multi-scale features from a bottlenecked, noised summary of them. We optimize with AdamW (Loshchilov & Hutter, 2019) for up to 150 epochs. Further details on the dataset and training can be found in §F and §G [1]

Baselines. We compare the following settings. (1) *RD* (Guo et al., 2025) trains reverse distillation directly on $\mathcal{U}_{\text{train}}$ without any correction, implicitly treating every image as normal regardless of

[1] Our code is available at the anonymous GitHub repository.

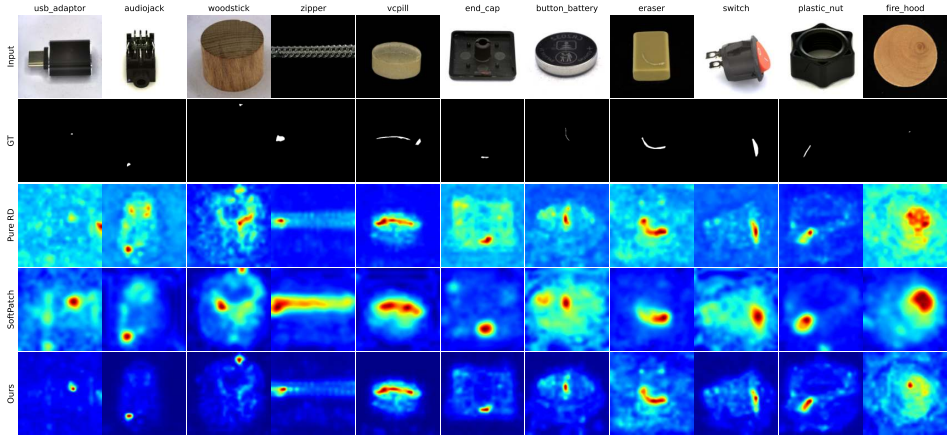


Figure 2: **Qualitative comparison across categories.** Anomaly maps for RD, SoftPatch, and Ours on anomalous test samples, alongside the ground-truth defect mask. Our method concentrates error on the true defect more tightly than RD or SoftPatch, both of which spread error over a larger, less precise region.

the true contamination rate α. (2) *SoftPatch* (Jiang et al., 2022) is a memory-bank baseline from a different architectural family, robust to contamination by construction and using no labeled pool $\mathcal{P}$. (3) *Ours* is the objective described in §3. (4) *Oracle* uses ground-truth labels, unavailable in any real PU deployment, to remove $\mathcal{U}_{\text{train}}$'s hidden anomalies before training, then trains RD on the resulting contamination-free set. The oracle is a diagnostic reference for how much a correction mechanism could help if contamination were perfectly known, best read as a clean-data reference rather than a strict upper bound. Per-category results are reported in Tab. 1.

We additionally compare against two baselines reported in the supplementary (§H, Tab. 5). *Two-stage* is the natural "clean then train" alternative: train RD on the contaminated pool, rank $\mathcal{U}_{\text{train}}$ by that model's own anomaly score, discard the top-scoring samples, and retrain RD on the rest. *PUAE*: re-implements the objective of Takahashi et al. (2026) on our backbone. A more detailed description of our baselines can be found in §H.

4.2 Analysis of Results

Quantitative Evaluation. We report image-level and pixel-level AUROC, together with the per-region overlap (PRO) metric, on the held-out test pool $\mathcal{U}_{\text{test}}$ of unseen samples. Tab. 1 compares RD, SoftPatch, Oracle, and our method across 30 Real-IAD categories. RD is a strong baseline even under contamination, since its structure makes it hard for the decoder to memorize individual patterns, which is why it already produces good results when $\mathcal{U}_{\text{train}}$ is contaminated. Still, our method improves the performance consistently across every category on both image-AUROC and PRO (per paired Wilcoxon signed-rank tests over categories §B, Tab. 4), and is ahead of SoftPatch on the large majority of categories across every metric, with only a couple of exceptions on image-AUROC. Our approach also clearly outperforms both supplementary baselines, Two-stage RD and PUAE (§C, Tab. 5). The oracle strategy remains ahead on image-AUROC, confirming, as expected, that a clean normal set is still a superior choice in training when available. Note that on PRO, our method is slightly ahead of the oracle on a majority of categories, which is not a contradiction: the oracle trains on a smaller, decontaminated pool, without the labeled anomaly set $\mathcal{P}$. Our model, instead, trains on the full contaminated pool plus $\mathcal{P}$, seeing more total data and a more diverse set of hard-normal content, which can outweigh the oracle's advantage, especially for localization.

Qualitative Evaluation. Fig. 2 visualizes anomaly maps produced by selected baselines and our method, alongside the ground-truth defect mask, on the contaminated pool $\mathcal{U}_{\text{test}}$ (Fig. 3 in §D provides more examples per category, including normal samples from the contaminated pool). RD's output is comparatively diffuse across the whole object. SoftPatch's error map is larger and more blurred, tending to spread the error beyond the true defect mask. Our approach concentrates error on

Table 2: **Ablation study.** Contribution of each loss term. *Ours* is the full objective.

Mode	AUC-Image	PRO	P-gap
RD	0.847	0.934	0.0069
$\mathcal{L}_U$	0.861	0.947	0.0082
$\mathcal{L}_U + \mathcal{L}_{P,\text{anchor}}$	0.863	0.949	0.0071
$\mathcal{L}_U + \mathcal{L}_{P,\text{hinge}}$	0.917	0.956	0.0643
$\mathcal{L}_U + \mathcal{L}_{P,\text{hinge}} + \mathcal{L}_{P,\text{anchor}}$	**0.921**	**0.958**	**0.0330**
Oracle	0.929	0.951	0.0070

the true defect for anomalies while staying quieter on normals, with the resulting high-error regions tending to coincide with the true defect location despite never receiving pixel-level supervision. Note that since our method builds on the same RD backbone, its error maps remain correlated with RD's, but the hinge and anchor terms sharpen the response into a more peaked signal around the defect and suppress background noise elsewhere, giving a tighter, more precise localization boundary.

Otsu separation thresholds. We further analyze what the Otsu split represents. Fig. 4 shows qualitative examples of how Otsu-flagged regions correspond to the GT anomaly mask for different categories. Even without labels as a supervisory signal, the split on the contaminated set $\mathcal{U}_{\text{test}}$ tracks hidden anomaly status at the level of region size: the flagged region is usually smaller on parts with hidden anomalies $\mathcal{U}_{\text{anom}}$ than on the normal ones $\mathcal{U}_{\text{norm}}$, so in anomalous images their error is concentrated into a tighter region while in normal images it is more spread.

We support this finding quantitatively in Tab. 6 which reports the per-image flagged ratio, precision, and recall of the flagged region $F(\boldsymbol{x})$ against the ground-truth defect mask (used here only for this diagnostic, never for supervision), wherever one exists, to check what the Otsu split actually captures. On both anomalous sets, labeled $\mathcal{P}$ and hidden $\mathcal{U}_{\text{anom}}$, recall is high: the flagged region rarely misses the defect (since the flagged fraction stays substantially smaller than the whole image on average). Precision is low, partly because Otsu has no notion of the anomaly's true extent, and partly because real defects are often small.

4.3 Ablation Study

Loss terms. To isolate the contribution of each term in the full objective Eq. (4), we ablate the full objective. $\mathcal{L}_{\mathcal{U}}$ keeps only the push-down term on $\mathcal{U}$. $\mathcal{L}_{\mathcal{P},\text{anchor}}$ and $\mathcal{L}_{\mathcal{P},\text{hinge}}$ add the anchor and hinge terms on $\mathcal{P}$ individually, without the other. Tab. 2 reports Image-AUROC, PRO, and the P-side overfitting gap for each setting: ${R^+}_{\text{train@5}}(\mathcal{P}) - {R^+}_{\text{val@5}}(\mathcal{P})$, the top-5%-largest-pixel risk statistic computed on the labeled training set $\mathcal{P}_{\text{train}}$ minus the same statistic on a disjoint validation set $\mathcal{P}_{\text{val}}$. A larger gap indicates a stronger tendency to fit the specific labeled examples rather than defects in general.: ${R^+}_{\text{train@5}}(\mathcal{P}) - {R^+}_{\text{val@5}}(\mathcal{P})$, the top-5%-largest-pixel risk statistic computed on the labeled training set $\mathcal{P}_{\text{train}}$ minus the same statistic on a disjoint validation set $\mathcal{P}_{\text{val}}$. A larger gap indicates a stronger tendency to fit the specific labeled examples rather than defects in general.

The $\mathcal{L}_{\mathcal{U}}$ term alone already improves over RD despite seeing no labeled anomalies at all, showing that the per-image Otsu push-down extracts useful signal in isolation. Adding an anchor on top of $\mathcal{L}_{\mathcal{U}}$ barely moves detection, confirming that the anchor's role is not separation: it only pulls $\mathcal{P}$'s non-flagged region down, so it has no mechanism to push a defect region's error up. Hinge is the term that drives detection: $\mathcal{L}_{\mathcal{U}} + \mathcal{L}_{\mathcal{P},\text{hinge}}$ closes most of the remaining gap to the full model on Image-AUROC in every category. But this comes at a cost: the P-gap widens substantially, i.e., hinge alone lets the decoder overfit the small labeled pool $\mathcal{P}$. Adding anchor back (the full objective) repairs this. Hinge and anchor are therefore complementary: hinge drives separation, anchor keeps the small labeled pool from being exploited into an overfit solution.

Alternative hinge form. The hinge term in Eq. (4) can be written in two ways, depending on where the margin is applied relative to the average over $\mathcal{P}$:

$$\mathcal{L}^{\text{pooled}}_{\mathcal{P},\text{hinge}} = \max\big(0, m - \mathbb{E}_{\boldsymbol{x}\in\mathcal{P}}[R^+(\boldsymbol{x})]\big), \tag{5}$$

$$\mathcal{L}^{\text{per-sample}}_{\mathcal{P},\text{hinge}} = \mathbb{E}_{\boldsymbol{x}\in\mathcal{P}}\big[\max\big(0, m - R^+(\boldsymbol{x})\big)\big]. \tag{6}$$

Table 3: Batch-level (pooled) vs. per-sample hinge.

	AUC-I	AUC-P	PRO	P-gap
Pooled	0.923	0.994	0.946	0.0527
Per-sample	**0.928**	0.994	**0.956**	**0.0329**

By Jensen's inequality, $\mathbb{E}[\max(0, m - R^+(\boldsymbol{x}))] \geq \max(0, m - \mathbb{E}[R^+(\boldsymbol{x})])$. The pooled form can therefore reach zero loss even when individual examples remain poorly separated, as long as others in the same batch compensate on average. Tab. 3 reports Image-AUROC/Pixel-AUROC/PRO alongside the same P-gap diagnostic introduced above. A large gap means that the hinge term has raised reconstruction error on the specific images it trains on without that effect transferring to unseen anomalies, which means memorization rather than generalization. The model trained on the per-sample hinge term achieves better performance than its counterpart.

5 Conclusions and Future Work

In this work, we introduced a positive-unlabeled methodology for anomaly detection that operates at the pixel level. In particular, we have devised a variant of the classic PUL setting for the localization of the anomaly within an image, and have applied the training procedure to a reverse-distillation architecture, operating at the feature space, to obtain a per-pixel anomaly score map.

The experiments confirm the superior performance against the other models. On the Real-IAD dataset, our PUL objective consistently improves over an uncorrected reverse-distillation and a memory-bank baseline, on both image-level detection and pixel-level localization, and closes most of the gap to an oracle trained on decontaminated data, all without ever knowing the real contamination rate in the data. Additionally, our ablation study shows that the terms introduced in the loss all bring a significant improvement to the performance, confirming the soundness of our developed methodology.

Our PUL setting assumes that labeled positives are a uniform random sample from P_a, independent of $\boldsymbol{x}$ – the standard SCAR assumption (Elkan & Noto, 2008; Bekker & Davis, 2020). Without it, $\mathcal{P}$ would carry no information about the hidden anomalies in $\mathcal{U}$. As future work, we plan to relax this assumption through active learning, implementing an adaptive PUL-AD approach that learns from a $\mathcal{P}$ which is not fixed during training, but can instead be updated with new anomalous samples.

References

Anish Acharya, Sujay Sanghavi, Li Jing, Bhargav Bhushanam, Dhruv Choudhary, Michael Rabbat, and Inderjit Dhillon. Positive unlabeled contrastive learning. *arXiv preprint arXiv:2206.01206*, 2022.

Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael Rabbat, Yann LeCun, and Nicolas Ballas. Self-supervised learning from images with a joint-embedding predictive architecture. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15619–15629, 2023.

Adrien Bardes, Quentin Garrido, Jean Ponce, Xinlei Chen, Michael Rabbat, Yann LeCun, Mahmoud Assran, and Nicolas Ballas. Revisiting feature prediction for learning visual representations from video. *Transactions on Machine Learning Research*, 2024.

Jessa Bekker and Jesse Davis. Learning from positive and unlabeled data: a survey. *Machine Learning*, 109(4), 2020.

Xuxi Chen, Wuyang Chen, Tianlong Chen, Ye Yuan, Chen Gong, Kewei Chen, and Zhangyang Wang. Self-PU: Self boosted and calibrated positive-unlabeled training. In Hal Daumé III and Aarti Singh (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 1510–1519. PMLR, 2020.

Hanqiu Deng and Xingyu Li. Anomaly detection via reverse distillation from one-class embedding. In *2022 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*. IEEE, 2022.

Marthinus Christoffel du Plessis, Gang Niu, and Masashi Sugiyama. Convex formulation for learning from positive and unlabeled data. In Francis Bach and David Blei (eds.), *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 1386–1394. PMLR, 2015.

Charles Elkan and Keith Noto. Learning classifiers from only positive and unlabeled data. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 213—-220. Association for Computing Machinery, 2008.

Jia Guo, Shuai Lu, Weihang Zhang, Fang Chen, Huiqi Li, and Hongen Liao. Dinomaly: The less is more philosophy in multi-class unsupervised anomaly detection. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20405–20415. IEEE, 2025.

Xi Jiang, Jianlin Liu, Jinbao Wang, Qiang Nie, Kai Wu, Yong Liu, Chengjie Wang, and Feng Zheng. SoftPatch: Unsupervised anomaly detection with noisy data. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 15433–15445. Curran Associates, Inc., 2022.

Ryuichi Kiryo, Gang Niu, Marthinus Christoffel du Plessis, and Masashi Sugiyama. Positive-unlabeled learning with non-negative risk estimator. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Aodong Li, Chen Qiu, Marius Kloft, Padhraic Smyth, Stephan Mandt, and Maja Rudolph. Deep anomaly detection under labeling budget constraints. In *International Conference on Machine Learning*, pp. 19882–19910. PMLR, 2023.

Zhikang Liu, Yiming Zhou, Yuansheng Xu, and Zilei Wang. SimpleNet: A simple network for image anomaly detection and localization. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20402–20411. IEEE, 2023.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mido Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jégou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision. *Trans. Mach. Learn. Res.*, 2024, 2024.

Nobuyuki Otsu. A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 9(1):62–66, 1979.

Chen Qiu, Aodong Li, Marius Kloft, Maja Rudolph, and Stephan Mandt. Latent outlier exposure for anomaly detection with contaminated data. In *International conference on machine learning*, pp. 18153–18167. PMLR, 2022.

Karsten Roth, Latha Pemula, Joaquin Zepeda, Bernhard Schölkopf, Thomas Brox, and Peter Gehler. Towards total recall in industrial anomaly detection. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2022.

Lukas Ruff, Robert Vandermeulen, Nico Görnitz, Lucas Deecke, Shoaib Ahmed Siddiqui, Alexander Binder, Emmanuel Müller, and Marius Kloft. Deep one-class classification. In Jennifer Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 4393–4402. PMLR, 2018.

Lukas Ruff, Robert A. Vandermeulen, Nico Görnitz, Alexander Binder, Emmanuel Müller, Klaus-Robert Müller, and Marius Kloft. Deep semi-supervised anomaly detection. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020.

Hiroshi Takahashi, Tomoharu Iwata, Atsutoshi Kumagai, and Yuuki Yamanaka. Deep positive-unlabeled anomaly detection for contaminated unlabeled data. *Neurocomputing*, 2026.

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

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

Botai Yuan, Chen Gong, Dacheng Tao, and Jie Yang. Weighted contrastive learning with hard negative mining for positive and unlabeled learning. *IEEE Transactions on Neural Networks and Learning Systems*, 36(6):10515–10529, 2025.

Yunrui Zhao, Qianqian Xu, Yangbangyan Jiang, Peisong Wen, and Qingming Huang. Dist-PU: Positive-unlabeled learning from a label distribution perspective. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14441–14450. IEEE, 2022.

A Derivation of Eq. (3)

Theorem A.1. *Let* $\alpha \in [0, 1)$ *and let*

$$P_u(\boldsymbol{x}) = (1-\alpha)P_n(\boldsymbol{x}) + \alpha P_a(\boldsymbol{x}) \tag{7}$$

hold. Then:

$$\mathbb{E}_{P_n}[R^+(\boldsymbol{x})] = (1-\alpha)^{-1}\big(\mathbb{E}_{P_u}[R^+(\boldsymbol{x})] - \alpha\mathbb{E}_{P_a}[R^+(\boldsymbol{x})]\big). \tag{8}$$

Proof (sketch). The identity follows from standard probability and measure theory results, and holds for every bounded Borel-measurable function[2] f, in particular for $f = R^+$.

First, note that P_u, P_n, P_a are densities which can be traced back to a common reference measure μ (which always exists). Let A be a Borel set. Integrating Eq. (7) over A with respect to μ gives:

$$\begin{aligned} P_u(\boldsymbol{x}) &= (1-\alpha)P_n(\boldsymbol{x}) + \alpha P_a(\boldsymbol{x}) && \Longrightarrow \\ \int_{\boldsymbol{x}\in A} P_u(\boldsymbol{x})\mathrm{d}\mu &= (1-\alpha)\int_{\boldsymbol{x}\in A} P_n(\boldsymbol{x})\mathrm{d}\mu + \alpha\int_{\boldsymbol{x}\in A} P_a(\boldsymbol{x})\mathrm{d}\mu && \Longrightarrow \\ \mathbb{E}_{P_u}[\mathbf{1}_A] &= (1-\alpha)\mathbb{E}_{P_n}[\mathbf{1}_A] + \alpha\mathbb{E}_{P_a}[\mathbf{1}_A], \end{aligned} \tag{9}$$

and since $\alpha < 1$,

$$\mathbb{E}_{P_n}[f] = (1-\alpha)^{-1}\big(\mathbb{E}_{P_u}[f] - \alpha\mathbb{E}_{P_a}[f]\big) \tag{10}$$

holds for $f = \mathbf{1}_A$. Further, by linearity of the expected value, the result holds for every f that is a *simple function*, *i.e.*, a linear combination of indicator functions. The result can be further extended to any f bounded and Borel-measurable, which is just the uniform limit of simple functions.

It is sufficient, then, to prove that $R^+(\boldsymbol{x})$ is a bounded Borel-measurable function, which, in our case, is true since $R^+(\boldsymbol{x}) \in [0, 2]$, and it is obtained by composition of Borel-measurable functions of $\boldsymbol{x}$ (the RD model + Otsu's thresholding).

□

B Significance Tests

Table 4: **Statistical significance.** Paired Wilcoxon signed-rank tests of Ours against each other method over the 30 categories. Δ: mean difference (Ours − other). W/T/L: number of categories where Ours is better / tied / worse. Bold: $p < 0.0056$ (Bonferroni, 9 tests).

Ours vs.	Metric	Δ	W/T/L	p
RD	AUC-I	$+0.058$	30/0/0	$\mathbf{1.7E{-}6}$
	AUC-P	$+0.003$	22/6/2	$\mathbf{3.0E{-}4}$
	PRO	$+0.024$	29/0/1	$\mathbf{2.6E{-}6}$
SoftPatch	AUC-I	$+0.036$	27/1/2	$\mathbf{3.1E{-}5}$
	AUC-P	$+0.009$	26/2/2	$\mathbf{7.3E{-}5}$
	PRO	$+0.038$	27/1/2	$\mathbf{3.6E{-}5}$
Oracle	AUC-I	-0.014	5/0/25	$\mathbf{2.6E{-}4}$
	AUC-P	-0.002	5/15/10	$5.6E{-}2$
	PRO	$+0.007$	21/2/7	$5.6E{-}3$

To check that the improvements in Tab. 1 are not driven by a few categories, we run paired two-sided Wilcoxon signed-rank tests over the 30 categories, comparing our method against each other method (Tab. 4). With Bonferroni correction for the nine tests (3 baselines × 3 metrics), the significance threshold is $p < 0.05/9 \approx 0.0056$. Our method is significantly better than RD and SoftPatch on all three metrics. Against the Oracle, Ours is significantly worse on image-AUROC, statistically indistinguishable on pixel-AUROC (mostly ties), and better on PRO (borderline after Bonferroni correction).

Table 5: **Comparison against PUAE and Two-stage.** Test-set Image-AUROC, Pixel-AUROC, and PRO on the Real-IAD categories where PUAE (Takahashi et al., 2026) was evaluated, against RD, Two-stage, and Ours. Bold marks the best of the four methods per column.

		RD			Two-stage RD			PUAE			Ours		
Category	α	AUC-I	AUC-P	PRO	AUC-I	AUC-P	PRO	AUC-I	AUC-P	PRO	AUC-I	AUC-P	PRO
audiojack	0.212	0.863	0.995	0.951	0.861	0.993	0.936	0.740	0.905	0.073	**0.920**	**0.997**	**0.959**
bottle_cap	0.279	0.854	0.994	0.960	0.765	0.837	0.453	0.923	0.911	0.047	**0.958**	**0.999**	**0.980**
eraser	0.248	0.902	0.997	0.950	0.908	0.997	0.943	0.838	0.931	0.184	**0.939**	**0.998**	**0.972**
fire_hood	0.226	0.824	0.996	0.902	0.754	0.988	0.792	0.790	0.873	0.000	**0.911**	**0.998**	**0.946**
mint	0.353	0.721	0.975	0.800	0.676	0.843	0.314	0.698	0.798	0.026	**0.803**	**0.989**	**0.880**
plastic_nut	0.206	0.907	0.998	0.976	0.901	0.998	0.964	0.891	0.881	0.000	**0.940**	**0.999**	**0.984**
rolled_strip_base	0.427	0.933	**0.999**	0.982	0.971	0.998	0.966	0.950	0.783	0.000	**0.981**	**0.999**	**0.992**
tape	0.242	0.945	0.998	0.972	0.960	0.998	0.961	**0.976**	0.907	0.001	0.974	**0.999**	**0.977**
transistor1	0.359	0.853	0.989	0.950	0.822	0.976	0.872	0.863	0.769	0.003	**0.964**	**0.993**	**0.975**
usb_adaptor	0.255	0.834	0.997	0.944	0.825	0.994	0.940	0.724	0.860	0.001	**0.898**	**0.998**	**0.951**
vcpill	0.246	0.872	0.989	0.902	0.742	0.944	0.753	0.819	0.920	0.044	**0.927**	**0.991**	**0.930**
zipper	0.392	0.978	0.985	0.946	0.983	**0.993**	0.945	0.987	0.768	0.000	**0.988**	0.991	**0.966**
Mean		0.874	0.993	0.936	0.847	0.963	0.820	0.850	0.859	0.032	**0.934**	**0.996**	**0.959**

C Additional Quantitative Results

Aside from the baselines reported in the main paper, we compare our method against two more baselines: *Two-stage RD* where we train RD on the contaminated pool, then discard the top-scoring samples and retrain RD on the rest; *PUAE* which re-implements the PUL objective of Takahashi et al. (2026) on our backbone.

D Additional Qualitative Results

In Fig. 3, we provide additional examples per category for the qualitative analysis.

E Otsu's thresholding: qualitative and quantitative evaluation

In Fig. 4 and Tab. 6, we report the quantitative and qualitative results of Otsu's thresholding analysis, as discussed in §4.2

F Dataset details

Real-IAD provides 5 images (camera angles) of each physical part. We call such a part an *instance*. We split the dataset at the level of instances: every instance belongs to exactly one pool, so images of the same part never appear in two pools. This rules out leakage between training, validation, and test, and ensures that the labeled anomalies in $\mathcal{P}$ come from different instances than the hidden anomalies in $\mathcal{U}$.

In particular, for each category, we reserve a small subset of anomalous instances as the labeled pool $\mathcal{P}$, split into a training portion $\mathcal{P}_{\text{train}}$ and a disjoint validation portion $\mathcal{P}_{\text{val}}$ used for the overfitting diagnostic. Only the views where the defect is annotated are kept for these instances, and their remaining views are discarded and not used anywhere. The remaining instances form the unlabeled pool $\mathcal{U}$, which we further split into $\mathcal{U}_{\text{train}}, \mathcal{U}_{\text{val}}, \mathcal{U}_{\text{test}}$. All splits are additionally stratified by defect subtype, so that every pool contains a representative mixture rather than being skewed toward the most common type (SCAR). The resulting contamination rate α varies naturally across categories rather than being chosen by us in advance (Tab. 7). Ground-truth labels and image masks are retained only for diagnostics and final evaluation, never for training. Splits are generated from a fixed random seed for reproducibility.

[2]The result is further extensible to unbounded functions, but it is out of scope in the context of this work.

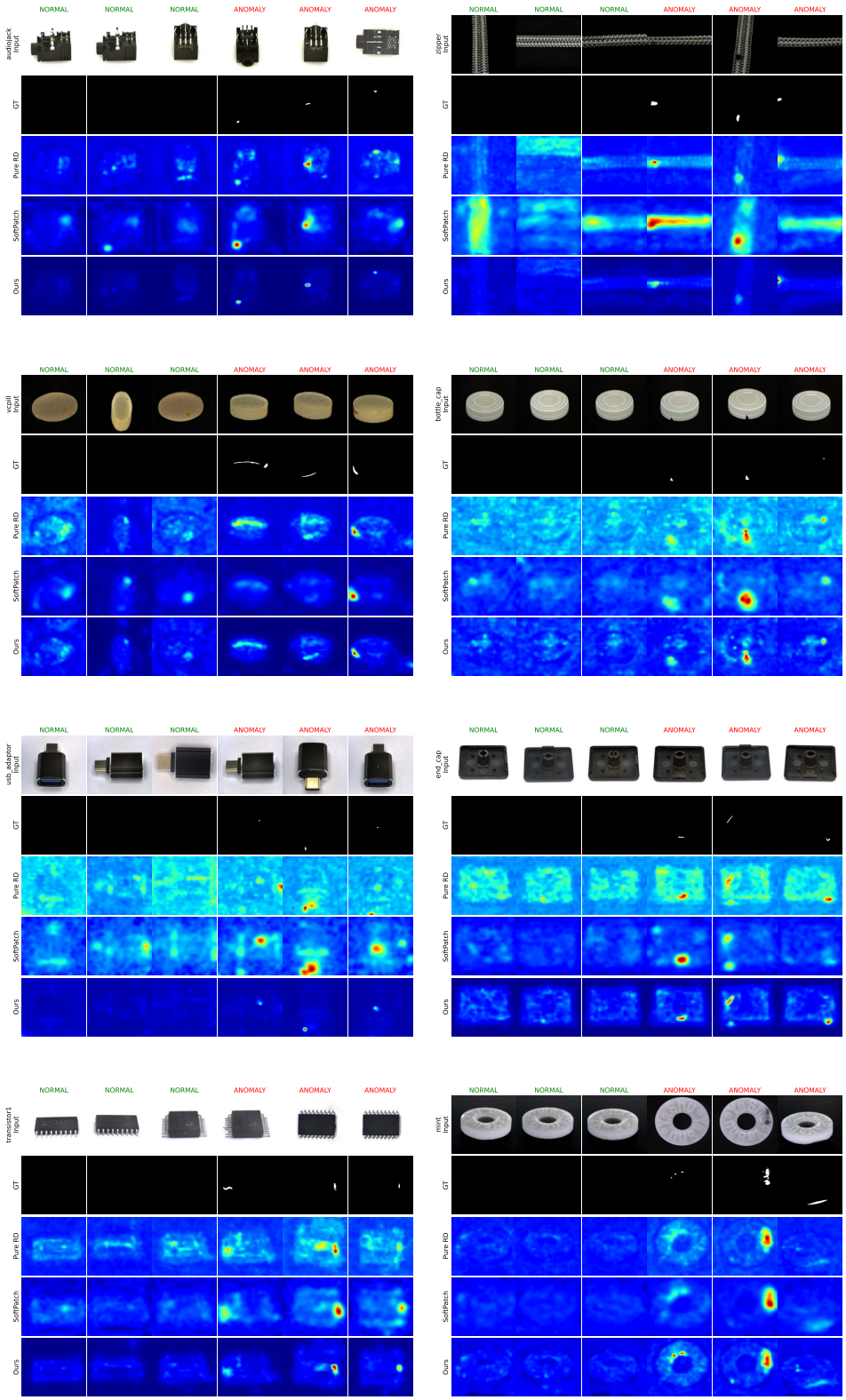


Figure 3: **Extended qualitative comparison.** Anomaly maps on the unseen contaminated test pool for RD, SoftPatch and Our method. Columns are individual images (normal, then anomalous); rows are, top to bottom, the input, the ground-truth defect mask (blank for normal images), and error maps of methods. Ours concentrates on the true anomalies more precisely and stays low on normals.

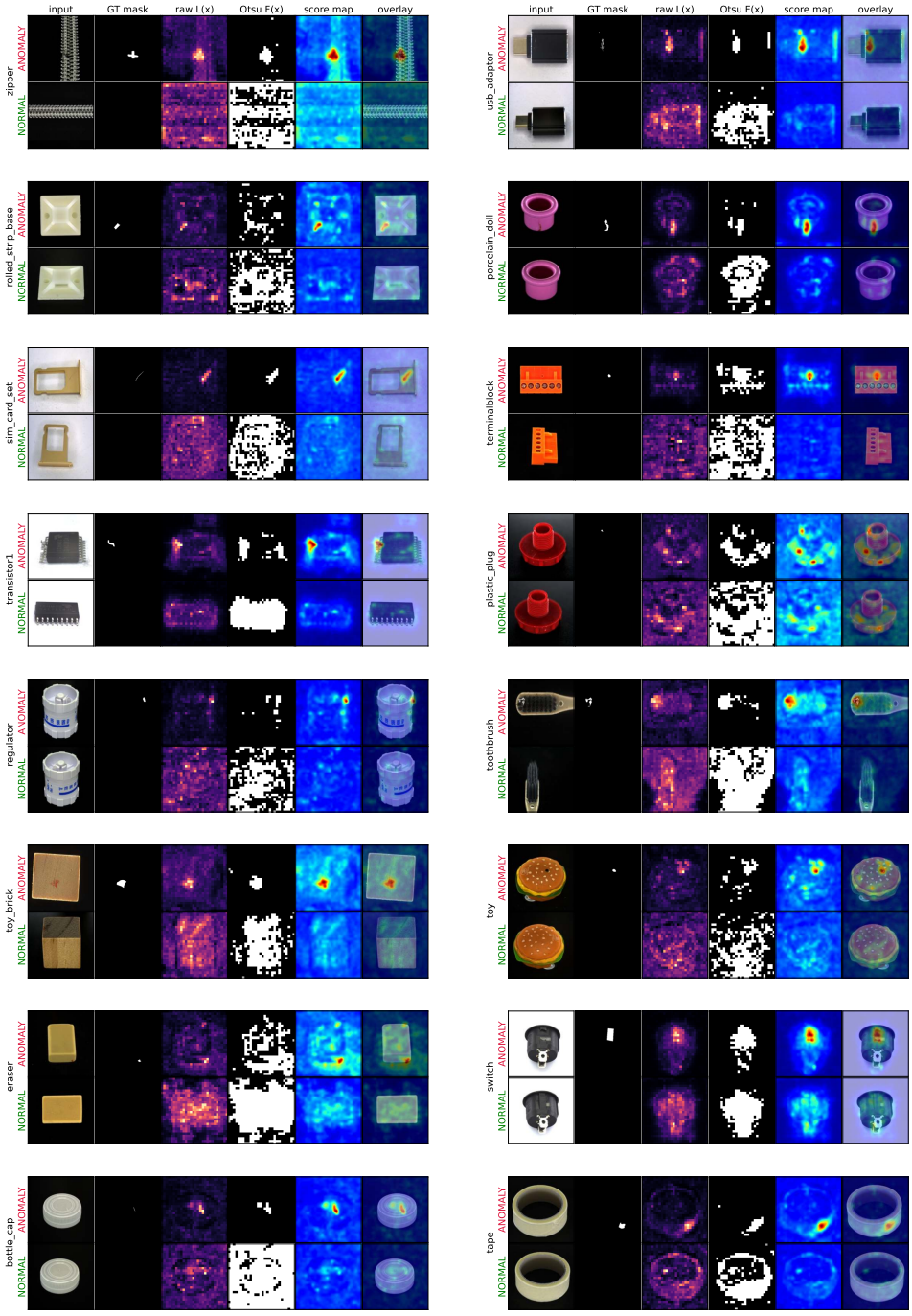


Figure 4: **Otsu split visualization.** Qualitative examples per category, drawn from $\mathcal{U}_{\text{test}}$: input, GT mask, the raw error map $\mathcal{E}(\boldsymbol{x})$, the Otsu-flagged region $F(\boldsymbol{x})$, and the resulting score map (with overlay), for one hidden anomalous (top row) and one hidden normal (bottom row) example. The hidden normal/anomalous label and the GT mask are never used for supervision, only for diagnostics.

Table 6: **Otsu split analysis.** Per-image Otsu-flagged fraction, precision, and recall against the real GT defect mask, per-sample-hinge full model. $\mathcal{U}_{\text{norm}}$/$\mathcal{U}_{\text{anom}}$ split $\mathcal{U}_{\text{test}}$ by its hidden ground-truth label; precision/recall are only defined where a real defect mask exists ($\mathcal{P}$ and $\mathcal{U}_{\text{anom}}$).

	$\mathcal{P}$			$\mathcal{U}$ (all)	$\mathcal{U}_{\text{norm}}$	$\mathcal{U}_{\text{anom}}$		
Category	fraction	precision	recall	fraction	fraction	fraction	precision	recall
audiojack	0.223	0.051	0.920	0.285	0.296	0.242	0.039	0.948
bottle_cap	0.270	0.038	0.934	0.565	0.702	0.239	0.037	0.981
button_battery	0.333	0.083	0.985	0.381	0.406	0.335	0.085	0.963
end_cap	0.382	0.026	0.981	0.457	0.482	0.416	0.013	0.987
eraser	0.350	0.081	0.953	0.550	0.629	0.319	0.075	0.966
fire_hood	0.296	0.019	0.985	0.378	0.401	0.302	0.016	0.992
mint	0.347	0.031	0.952	0.451	0.520	0.325	0.027	0.930
mounts	0.258	0.061	0.958	0.474	0.535	0.268	0.078	0.955
pcb	0.317	0.065	0.969	0.375	0.393	0.349	0.025	0.991
phone_battery	0.188	0.075	0.962	0.346	0.406	0.176	0.065	0.981
plastic_nut	0.320	0.053	0.972	0.517	0.581	0.307	0.054	0.967
plastic_plug	0.360	0.042	0.949	0.473	0.511	0.369	0.047	0.956
porcelain_doll	0.263	0.075	0.957	0.383	0.407	0.290	0.057	0.957
regulator	0.171	0.079	0.949	0.414	0.462	0.188	0.102	0.920
rolled_strip_base	0.160	0.074	0.948	0.337	0.466	0.166	0.071	0.945
sim_card_set	0.198	0.103	0.950	0.508	0.679	0.182	0.099	0.936
switch	0.257	0.067	0.933	0.354	0.405	0.257	0.070	0.966
tape	0.206	0.072	0.972	0.548	0.666	0.179	0.076	0.965
terminalblock	0.168	0.105	0.971	0.429	0.545	0.195	0.091	0.959
toothbrush	0.320	0.062	0.956	0.401	0.445	0.310	0.050	0.971
toy	0.283	0.032	0.916	0.391	0.440	0.308	0.027	0.939
toy_brick	0.437	0.039	0.960	0.489	0.508	0.442	0.015	0.989
transistor1	0.269	0.084	0.966	0.369	0.424	0.269	0.078	0.953
u_block	0.233	0.113	0.939	0.524	0.585	0.259	0.117	0.950
usb	0.306	0.050	0.977	0.389	0.423	0.325	0.042	0.970
usb_adaptor	0.148	0.057	0.884	0.415	0.485	0.195	0.043	0.953
vcpill	0.254	0.118	0.904	0.387	0.422	0.275	0.083	0.942
wooden_beads	0.316	0.033	0.976	0.386	0.405	0.339	0.030	0.982
woodstick	0.424	0.030	0.967	0.517	0.537	0.419	0.053	0.970
zipper	0.226	0.127	0.930	0.410	0.522	0.238	0.096	0.953
Mean	0.276	0.065	0.953	0.430	0.490	0.283	0.059	0.961

Table 7: **Dataset split statistics.** Real-IAD categories and our PU split. Pool sizes are given in number of images (views). α_{train}, α_{val}, α_{test} are the percentages of anomalous images in $\mathcal{U}_{\text{train}}$, $\mathcal{U}_{\text{val}}$ and $\mathcal{U}_{\text{test}}$.

Category	$\mathcal{P}_{\text{train}}$	$\mathcal{P}_{\text{val}}$	$\mathcal{U}_{\text{train}}$	$\mathcal{U}_{\text{val}}$	$\mathcal{U}_{\text{test}}$	α_{train} (%)	α_{val} (%)	α_{test} (%)
audiojack	229	141	3 075	650	650	21.0	22.3	20.9
bottle_cap	276	143	3 030	655	655	27.8	27.2	29.5
button_battery	382	191	3 025	650	650	36.6	37.5	34.6
end_cap	379	190	3 035	650	650	37.5	35.5	37.5
eraser	289	123	3 030	655	655	24.7	24.4	25.6
fire_hood	242	123	3 040	655	655	22.4	23.5	23.1
mint	380	198	3 015	645	645	35.6	34.0	35.5
mounts	235	128	3 055	660	660	21.8	24.1	22.6
pcb	388	215	3 035	650	650	38.6	39.2	40.2
phone_battery	275	126	3 025	650	650	26.9	25.4	26.0
plastic_nut	243	109	3 035	650	650	19.5	23.1	23.4
plastic_plug	252	128	3 020	650	650	23.8	22.5	26.8
porcelain_doll	205	99	3 035	645	645	21.9	23.9	20.6
regulator	220	85	3 065	655	655	17.7	18.2	17.6
rolled_strip_base	439	220	3 055	660	660	42.5	43.0	42.9
sim_card_set	367	179	3 045	655	655	35.2	34.5	34.4
switch	362	160	3 055	660	660	33.6	33.2	34.5
tape	243	129	3 105	660	660	24.4	23.5	24.1
terminalblock	381	159	3 035	645	645	33.9	34.7	33.2
toothbrush	353	182	3 040	645	645	35.0	34.1	32.2
toy	393	192	3 040	655	655	36.8	37.1	36.9
toy_brick	251	141	3 045	645	645	26.2	24.7	28.8
transistor1	379	186	3 025	650	650	36.4	34.3	35.1
u_block	237	113	3 015	650	650	20.1	20.6	18.9
usb	290	148	3 055	660	660	28.7	31.7	34.7
usb_adaptor	292	140	2 985	640	640	25.9	25.0	24.2
vcpill	259	140	3 040	655	655	24.7	24.6	24.1
wooden_beads	306	140	2 960	635	635	29.0	30.6	29.1
woodstick	193	100	3 020	655	655	17.9	19.7	16.8
zipper	423	212	3 030	650	650	39.2	38.6	39.4

G Implementation Details

We instantiate t and g_θ following the Dinomaly recipe (Guo et al., 2025). The teacher is a frozen, pretrained ViT-Base/14 variant of the DINOv2 encoder (Oquab et al., 2024) with registers, and the student is a compact trainable network that reconstructs the teacher's multi-scale features from a bottlenecked, noised summary of them. Unlike Dinomaly's original multi-class setting, we train a separate model per category to isolate the correction mechanism from cross-category effects. The inputs are resized to 392×392 (divisible by the patch size) at both training and test time. Teacher features are pooled into two fusion groups of four consecutive transformer blocks, mirrored by an eight-block linear-attention decoder in the same $4 + 4$ grouping. The bottleneck applies dropout with $p = 0.2$ as its noise source. We train with AdamW (Loshchilov & Hutter, 2019), learning rate $2E-4$, weight decay $1E-4$, batch size 16, for up to 150 epochs with early stopping on the held-out loss (evaluated on $\mathcal{U}_{\text{val}}$ and $\mathcal{P}_{\text{val}}$), computed without hidden $\mathcal{U}$-labels. The hyperparameters are set respectively to $m = 0.3$, $\beta = 0.3$, and $\gamma = 1$.

H Baselines

RD. RD is a Dinomaly-style (Guo et al., 2025) reverse distillation architecture, with a frozen DINOv2 teacher and a transformer-based student decoder, trained directly on $\mathcal{U}_{\text{train}}$ without any correction, using the hard-region mining $R(x)$. Its noised bottleneck, linear attention, and loose reconstruction constraint all work against the decoder trivially memorizing any single pattern, which is part of why it already produces reasonable anomaly maps even under contamination. This baseline implicitly treats every image as normal regardless of the true contamination rate α, and is the uncorrected baseline every other method in this comparison is measured against.

Two-stage RD. Two-stage is the natural "clean then train" alternative: train RD on the contaminated pool, rank $\mathcal{U}_{\text{train}}$ by that model's own anomaly score, discard the top-scoring images, and retrain RD on the rest. We give it the true per-category anomaly count, so it only has to rank the pool correctly, not also decide how many images to cut.

Its performance (Tab. 5) depends heavily on the quality of its own detect-then-clean filter. On categories where the initial RD-based ranking already separates anomalies well, retraining on the filtered pool is roughly neutral or a small improvement over RD. On categories where that ranking is poor, the filter removes mostly genuine hard normals rather than the hidden anomalies, and retraining on this now-worse pool drives performance well below RD, even though the filter is handed the true per-category anomaly count.

Oracle. Oracle uses ground-truth labels, unavailable in any real PU deployment, to remove $\mathcal{U}_{\text{train}}$'s hidden anomalies before training, then trains RD on the resulting contamination-free set (effectively a one-class model that never sees the labeled anomalies $\mathcal{P}$ at all). It is a diagnostic reference for how much a correction mechanism could help if contamination were perfectly known, not a baseline in the usual sense and not a strict upper bound on Ours either (Ours also uses $\mathcal{P}$, which Oracle never sees).

PUAE. To isolate the contribution of our spatial objective from that of the backbone, we compare it against the objective of Takahashi et al. (2026) under the same architecture: our frozen DINOv2 encoder and reverse-distillation decoder, rather than their original autoencoder reconstructing raw pixels, so that any difference reflects the correction mechanism rather than backbone choice. We give the loss the true contamination rate α for each category.

However, PRO collapses on every category (Tab. 5). The reason is direct: the term that treats $\mathcal{P}$'s score as anomalous always rewards a larger mean error on $\mathcal{P}$ and never stops, so training pushes it toward the ceiling; the term estimating the risk on normal data is smallest when the mean error on $\mathcal{U}$ equals α times the mean error on $\mathcal{P}$, so raising one drags up the other. Because the score is a mean over the whole image, the only way to raise it is to raise the error everywhere, not just at the defect, so nothing in the loss lets the model satisfy either term by degrading one region while leaving the rest untouched. The result is an error map that is uniformly high across the whole image, on both pools, with no local contrast left for localization to use.

SoftPatch. As a contamination-robust baseline from a different architectural family, we also compare against SoftPatch (Jiang et al., 2022): a memory-bank method built on a frozen, ImageNet-pretrained backbone. It constructs a coreset of normal-looking patch embeddings and down-weights patches whose local density marks them as likely contaminated. We feed it the same already-contaminated $\mathcal{U}_{\text{train}}$ split, in place of its own synthetic noise-injection protocol, and it uses no labeled pool $\mathcal{P}$ at all.

LOE. We do not report Latent Outlier Exposure (Qiu et al., 2022) as a quantitative baseline. Its score has no reconstruction or spatial analogue, so any adaptation to our backbone is a design choice rather than a faithful translation of the original method.

SOEL. We do not include active-learning methods, such as SOEL (Li et al., 2023), as a baseline. These methods grow $\mathcal{P}$ *during* training, querying an oracle for labels on samples selected by the model's own query strategy under an explicit labeling-budget schedule. Our setting instead fixes $\mathcal{P}$ once, before training begins, drawn independently of the model's training process. The question we study is how to make the best use of a small, already-fixed labeled set, not how to choose which examples are worth labeling.

I LLM USAGE

The authors used large language models to aid in polishing the writing of this manuscript. All scientific content, analyses, and conclusions are the authors' own, and the authors take full responsibility for the final text.