

INVERSE MEETS DISTILLATION: HETEROGENEOUS TEACHER–ASSISTANT DUAL-PATH LEARNING FOR UNSUPERVISED DEFECT DETECTION

Xin Zhan*, Yaqian Li, Ruihao Chen,
Wenming Zhang, Haibin Li

*Key Laboratory of Industrial Computer Control Engineering of Hebei Province,
Yanshan University, Qinhuangdao 066004, Hebei, China
Engineering Research Center of the Ministry of Education for Intelligent Control System
and Intelligent Equipment, Yanshan University, Qinhuangdao 066004, Hebei, China*

*E-mail: 18848828119@163.com

Submitted: 26th September 2025; Accepted: 29th January 2026

Abstract

Unsupervised defect detection is crucial for industrial inspection, but teacher–student (T–S) frameworks tend to overfit a single teacher’s feature manifold, leading to poor generalization on subtle anomalies. We introduce TAD++, a dual-path distillation framework that combines heterogeneous Teacher–Assistant–Student (T–A–S) guidance with a pseudo-defect inverse-distillation branch. A compact assistant, structurally distinct from the teacher, is trained to co-distill the student, thereby mitigating single-teacher bias. In parallel, the inverse-distillation path tasks the student with reconstructing normal appearances from defect-injected inputs, serving as a regularization term to prevent anomaly leakage. A dynamic attention weighting module adaptively fuses these heterogeneous guidance signals. Crucially, the assistant, inverse branch, and weight modules are strictly training-only. This design ensures that while TAD++ benefits from a rigorous multi-phase optimization, it maintains zero additional inference latency and memory overhead compared to standard T–S baselines. On MVTec AD, BTAD, and VisA, TAD++ achieves consistent improvements in both image-level detection and pixel-level localization, with extensive ablations confirming the efficacy of the heterogeneous dual-path design.

Keywords: unsupervised defect detection; dual-path distillation; Teacher–Assistant–Student; inverse distillation; dynamic attention

1 Introduction

Industrial products are ubiquitous in daily life, and defect detection is central to ensuring their reliability. In recent years, computer vision has gradually replaced labor-intensive manual inspection with deep learning systems that learn discriminative representations for defect detection and seg-

mentation. Related detection problems have also been investigated in other domains within this journal, such as electricity consumption monitoring and atrial fibrillation detection [1, 2].

In practice, however, normal samples are abundant while defective samples are scarce and costly to annotate. Purely supervised methods [3] there-

fore struggle with insufficient labels and high annotation costs. Semi/weakly supervised approaches partially relieve this burden by using a small set of labeled defects together with many normal samples, yet real-world industrial scenarios still suffer from the scarcity of defect exemplars, incomplete defect taxonomies, and large intra-class variability. Although data augmentation [4] can alleviate data scarcity, inappropriate transformations may introduce spurious patterns and even hurt performance [5]. Consequently, a prevailing trend is to learn only from normal data and treat deviations as anomalies, recasting defect detection as an unsupervised problem and substantially reducing labeling costs.

Teacher–student (T–S) frameworks have become a popular paradigm for unsupervised defect detection. Beginning with Bergmann et al. [6], knowledge is distilled from a teacher to a student so that both produce similar features on normal regions while exhibiting discrepancies on defects. The key challenge is to balance distillation in a way that preserves the student’s generalization: the student should mimic the teacher on normal samples yet avoid inheriting biases that suppress anomaly sensitivity. In practice, poor generalization often stems from overfitting the student to the teacher’s parameters or feature mapping. Multi-teacher variants [7] attempt to mitigate this by aggregating complementary teachers, but distribution mismatches between generic pretraining (e.g., ImageNet [8]) and target industrial data can persist, causing students to absorb features irrelevant to the normal target domain [9].

To address these issues, we propose Teacher–Assistant Distillation (TAD), building upon the AST architecture [10]. In TAD, a compact assistant is first trained under the guidance of the teacher on the target dataset and then serves as an additional, heterogeneous source of supervision for the student. This design diversifies guidance beyond a single teacher and reduces overfitting to one feature mapping. Moreover, we introduce a dynamic attention weight module that outputs a single-channel attention map to adaptively balance the contributions from the teacher and the assistant against the student’s self-learning signal. For pixel-level localization, we further integrate the teacher’s defect scores to refine segmentation.

We evaluate TAD on three public benchmarks,

MVTec AD [11], BTAD [12], and VisA [13]. Our method achieves average image-level AUROC of 99.35%, 96.12%, and 96.52%, and average pixel-level AUROC of 96.56%, 98.02%, and 98.99%, respectively, demonstrating competitive performance and robust localization.

Our main contributions are threefold:

- We propose a heterogeneous Teacher–Assistant Distillation framework in which a compact assistant, trained via outcome-based knowledge distillation, co-guides the student together with the teacher. This diversified supervision curbs single-teacher overfitting and strengthens anomaly sensitivity.
- We design a lightweight dynamic attention weight module that produces a single-channel attention map to adaptively balance dual-teacher guidance with the student’s independent learning; additionally, we inject the teacher’s defect scores into the pixel-level head to enhance localization.
- We validate TAD on MVTec AD, BTAD, and VisA, reporting strong image-level and pixel-level AUROC together with extensive ablations, which collectively confirm the effectiveness of the proposed components.

2 Related Work

2.1 Unsupervised Defect Detection: Taxonomy and Trends

Unsupervised defect detection methods can be broadly grouped into four families. Reconstruction-based approaches learn to reproduce normal appearances and detect anomalies via input–reconstruction discrepancies, typically using Autoencoders (AE) [14, 15], Variational Autoencoders (VAE) [16], and Generative Adversarial Networks (GAN) [17, 18]. Style-preserving or distribution-regularized designs (e.g., OE-SDN [19] and adversarially trained VAE [20]) mitigate content drift, while memory/feature banks [21, 22] and frequency-aware enhancement (AGMM) [23] aim to avoid over-generalization to defects. Inpainting/self-supervised methods synthesize pseudo defects and train networks to restore normal patterns [24–28], with recent strategies focusing on boundary-aware and progressive anomaly

synthesis [29]. One-class classification constructs compact normality regions in feature space (OC-SVM/SVDD [30, 31], Deep SVDD [32]) and learns distance-minimizing or transferable hypersphere objectives [33, 34]. Finally, feature distance methods encompass template matching [35, 36] and statistical modeling [37, 38]. While these streams achieve strong performance in various settings, most of them rely on a single model to fit normality; under distribution shifts or fine-grained localization requirements, single-model assumptions can limit robustness and sensitivity.

2.2 Teacher–Student Distillation for Anomaly Detection

Teacher–student (T–S) frameworks transfer normal knowledge so that teacher–student discrepancies highlight anomalies. Since [6], distillation signals have evolved from single-layer guidance to multi-scale matching [39], and have been combined with pseudo-defect generation to enhance segmentation sensitivity [40, 41]. A persistent challenge is overfitting the student to the teacher’s parameters or feature mapping, which can suppress anomaly sensitivity. Moreover, when teachers are pre-trained on generic datasets such as ImageNet [8], distribution mismatches with industrial targets may cause students to absorb features irrelevant to the normal target domain [9]. These issues motivate designs that diversify supervision and explicitly regularize student learning.

2.3 Multi-Teacher, Heterogeneous Designs, and Dynamic Weighting

Aggregating multiple teachers can mitigate single-teacher bias and improve adaptability [42–44], yet many approaches adopt static or equal weighting and may overlook contextual differences across samples. To address this, adaptive/dynamic weighting has been explored via latent factors and multi-level targets [7], meta-weight integration [45], reinforcement learning for teacher selection [46], and reconstruction-error-driven importance updates [47]. Orthogonally, heterogeneous T–S designs reduce direct parameter copying and induce feature variance, e.g., pairing a normalized-flow teacher with a feedforward student [10], exploiting reverse or bidirectional distillation [48, 49], or amplifying teacher–student (T–S) be-

havioral margins in structural similarity layers [50]. These lines collectively suggest that (i) diversified supervision and (ii) architectural heterogeneity are key to curbing overfitting while retaining anomaly sensitivity. Unlike the AST baseline [10], which relies solely on a symmetric Teacher–Student pair, TAD++ introduces architectural heterogeneity via the Assistant and enforces normality through the inverse-distillation branch. Furthermore, distinct from previous multi-teacher methods that typically employ static or global weighting strategies, our dynamic weighting module enables pixel-wise adaptation, allowing the student to selectively prioritize the most reliable guidance signal at each spatial location.

2.4 Pixel-Level Calibration and Pseudo-Defect/Pretext Signals

Beyond image-level decisions, accurate localization benefits from calibrated pixel-level cues. Injecting teacher-side scores or structural priors can refine student maps and enlarge normal–anomalous margins [50], and multi-scale correspondence improves spatial precision [39]. When available, pseudo-defect or restoration pretexts provide additional regularization for segmentation heads [24–29]. Frequency-aware enhancements, such as AGMM [23], and memory/feature storage [21, 22] further stabilize pixel-level responses under texture and illumination variations.

Our approach aligns with these trends by combining diversified supervision and heterogeneity, while introducing dynamic weighting and pixel-level score fusion to balance teacher guidance with the student’s independent learning signal.

3 Proposed Method

We propose TAD++, a dual-path framework built upon heterogeneous Teacher–Assistant–Student (T–A–S) learning. Path A keeps the original teacher–assistant–student distillation; Path B is a training-only pseudo-defect inverse-distillation branch that regularizes the student toward the normal manifold. A lightweight dynamic attention weight network outputs a pixel attention map m and global weights (α, β) to adaptively fuse dual-teacher guidance. Following AST [10], the teacher

is a flow-based model trained on normal features; all branches take as input the concatenation of EfficientNet-B5 features [51] and sinusoidal positional encoding [52].

TAD++ is trained in three phases on normal samples: Phase A learns the teacher; Phase B distill the assistant from the frozen teacher; Phase C train the student under dual-teacher guidance with dynamic weights. Additionally, Path B (Sec. 3.4 and Figure 5) injects a training-only inverse-distillation signal.

3.1 Heterogeneous Models

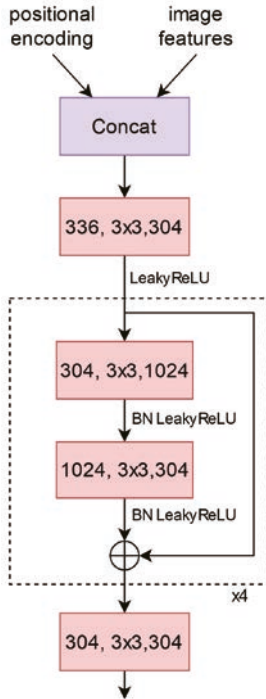


Figure 2. Student backbone: four residual 3×3 conv blocks ($304 \leftrightarrow 1024$) bracketed by 3×3 convs, preserving spatial resolution.

Student. As in Figure 2, the student has four identical residual blocks with two 3×3 convs (BN+LeakyReLU), preceded/followed by input/output conv layers. Each block expands then reduces channels, keeping the same channel size before/after the block.

Assistant. Figure 3 shows a shallow network with large receptive fields: three 6×6 conv+AvgPool blocks and a final 5×5 conv. Padding keeps spatial sizes; channels of intermediate layers equal the output channels.

Teacher. A bijective normalizing flow trained by maximum likelihood maps normal features to $\mathcal{N}(0, 1)$. The heterogeneity among $T/A/S$ reduces direct parameter copying and encourages independent learning.

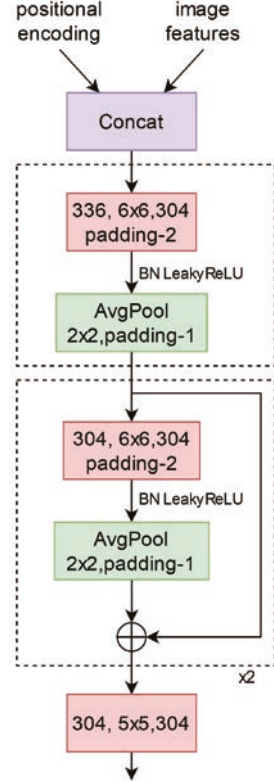


Figure 3. Assistant model: two 6×6 conv plus AvgPool macro blocks followed by a 5×5 conv forming a shallow large-receptive-field auxiliary teacher.

3.2 Dynamic Attention Weight Network

The dynamic attention head adaptively balances teacher vs. assistant guidance. Left branch concatenates teacher/assistant features to produce m ; right branch uses $\{F_T^4, F_A^4, F_S^4\}$ to compute (α, β) .

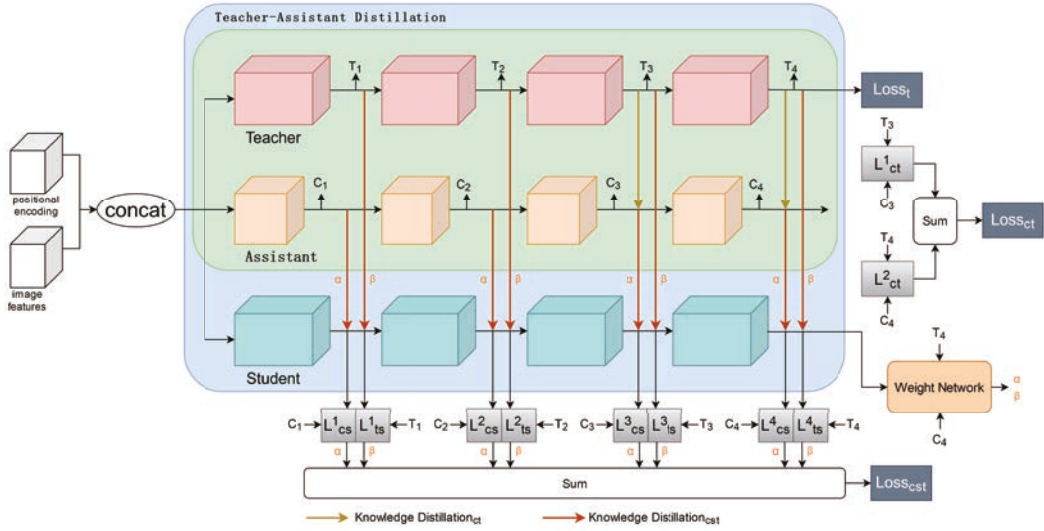


Figure 1. TAD++ training process. Phase A: teacher learning; Phase B: assistant distillation from the frozen teacher; Phase C: student dual-teacher distillation with dynamic weights. Per-scale losses L_{ct}^k , L_{ts}^k , L_{cs}^k are aggregated (“Sum”). Assistant and Weight Network are training-only; inference uses only the teacher–student discrepancy. See Figure 5 for Path B.

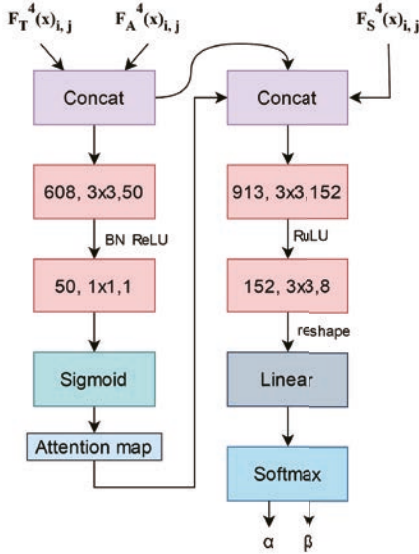


Figure 4. Dynamic Attention Weight Network. Left branch (as in [41]) produces a pixel attention map $m \in [0, 1]^{H \times W}$. Right branch outputs global weights (α, β) with $\alpha + \beta = 1$.

3.3 Path A: Dual-Teacher Distillation

At pyramid level $k \in \{1, 2, 3, 4\}$, denote feature vectors at location (i, j) by $F_T^k(\mathbf{x})_{i,j}$, $F_A^k(\mathbf{x})_{i,j}$, $F_S^k(\mathbf{x})_{i,j} \in \mathbb{R}^{C_k}$. The assistant is distilled from the teacher on the last two levels:

$$L_{AT} = \sum_{k=3}^4 \frac{1}{W_k H_k} \sum_{i,j} \|F_T^k(\mathbf{x})_{i,j} - F_A^k(\mathbf{x})_{i,j}\|_2^2. \quad (1)$$

During student learning, with pixel attention m and global weights (α, β) ,

$$D_{TS}^k = \frac{1}{W_k H_k} \sum_{i,j} m_{i,j} \|F_T^k(\mathbf{x})_{i,j} - F_S^k(\mathbf{x})_{i,j}\|_2^2, \quad (2)$$

$$D_{AS}^k = \frac{1}{W_k H_k} \sum_{i,j} m_{i,j} \|F_A^k(\mathbf{x})_{i,j} - F_S^k(\mathbf{x})_{i,j}\|_2^2. \quad (3)$$

Aggregating across scales,

$$L_{ATS} = \sum_{k=1}^4 (\alpha D_{TS}^k + \beta D_{AS}^k). \quad (4)$$

3.4 Path B: Pseudo-Defect Inverse Distillation

Beyond Path A, we generate pseudo defects from normal images via cut-and-paste, erasing, and boundary-aware synthesis [24–29].

Specifically, to ensure reproducibility, we generate pseudo-defects by selecting random rectangular regions with a scale range of $[0.02, 0.1]$ relative to the image area and applying corruption operations. The mixing ratio of normal to pseudo-defect samples in the training batch is set to $r = 0.5$. Given a normal input x and its pseudo-defect counterpart

$\tilde{x}$, the student (shared weights) is required to reconstruct the normal top-level feature from $\tilde{x}$:

$$L_{\text{rec}} = \frac{1}{WH} \sum_{i,j} \|F_S^4(\tilde{x})_{i,j} - F_S^4(x)_{i,j}\|_2^2. \quad (5)$$

This inverse-distillation regularizes the student to avoid anomaly leakage and complements structural regularization from heterogeneity.

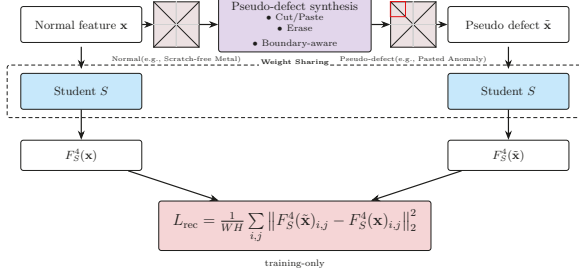


Figure 5. Path B: pseudo-defect inverse distillation. A normal feature x is corrupted to a pseudo-defect $\tilde{x}$ by cut/paste/erasing/boundary-aware synthesis. The student produces $F_S^4(\tilde{x})$ and $F_S^4(x)$; the reconstruction loss L_{rec} in Eq. (5) drives $\tilde{x}$ back to the normal manifold. This branch is training-only; at test time, only T and S are used.

3.5 Training Objectives and Optimization

Teacher objective (Phase A). Let the teacher map input features x to latent z ; its flow-based likelihood loss is

$$L_T = \frac{1}{WH} \sum_{i,j} \left[\frac{1}{2} \|z_{i,j}\|_2^2 - \log \left| \det \frac{\partial z_{i,j}}{\partial x_{i,j}} \right| \right]. \quad (6)$$

Alternating optimization (Phase C). We alternate two steps. The primary cycle updates the student using dual-teacher loss on normal batches and inverse-distillation loss on pseudo-defect batches (mixing ratio r):

$$\begin{aligned} \min_S \mathcal{L}_{\text{ATS}} \quad & \text{(normal batches),} \\ \min_S \mathcal{L}_{\text{rec}} \quad & \text{(pseudo-defect batches).} \end{aligned} \quad (7)$$

As summarized in Algorithm 1, we alternate a primary update of S and a secondary update of the weight head.

The secondary cycle updates only the weight head on hard pixels. Let

$$\mathcal{D} = \left\{ \|F_T^k(x)_{i,j} - F_S^k(x)_{i,j}\|_2^2, \|F_A^k(x)_{i,j} - F_S^k(x)_{i,j}\|_2^2 \mid k = 1, \dots, 4, (i, j) \right\}. \quad (8)$$

and set $h = \text{Quantile}(\mathcal{D}, q_{\text{hard}})$ with $q_{\text{hard}} \in [0, 1]$; backpropagate the weight head only on pixels with discrepancy $> h$.

Phase C total loss.

$$L_{\text{total}} = L_{\text{ATS}} + \gamma L_{\text{rec}}, \quad (9)$$

where γ balances dual-path signals; $(\lambda, \gamma, \eta, q_{\text{hard}}, r)$ are selected on a normal validation split and kept fixed.

Algorithm 1 HOEE algorithm

Require: Dynamic network $\mathcal{G} = \{G^{(1)}, G^{(2)}, \dots, G^{(t)}\}, \beta_1, \beta_2$

Ensure: $\{C^{(1)}, C^{(2)}, \dots, C^{(t)}\}$

- 1: $C^{(1)} \leftarrow \text{Louvain}(G^{(1)})$
 - 2: **for** $t = 2 : T$ **do**
 - 3: Construct harmonic motif adjacency matrix HMA from $A^{(t)}$ via Equation (10).
 - 4: Construct participation adjacency matrix PM from $HMA(t)$ via Equation (9).
 - 5: Compute HAP_i of every node i via Equation (8).
 - 6: $\Delta E \leftarrow$ a set of edges changed from $E^{(t-1)}$ to $E^{(t)}$.
 - 7: **for** $e_{ij} \in \Delta E$ **do**
 - 8: randomly generate a threshold: thres .
 - 9: **if** $\text{thres} \leq \beta_1$ **then**
 - 10: **if** $HAP_i < \beta_2$ and $HAP_j < \beta_2$ **then**
 - 11: continue
 - 12: **end if**
 - 13: **end if**
 - 14: $\text{active_edge.append}(e_{ij})$
 - 15: **end for**
 - 16: **for** $e_{ij} \in \text{active_edge}$ **do**
 - 17: $\text{hybrid_G} = G^{(t)} + \text{active_edge}$
 - 18: $\text{tri} = \text{detect_tri}(\text{hybrid_G})$
 - 19: **if** i in tri and j in tri **then**
 - 20: $\text{enhance_G}^{(t)}.add_edge(e_{ij})$
 - 21: **end if**
 - 22: **end for**
 - 23: $C^{(t)} \leftarrow \text{Louvain}(G^{(t)})$
 - 24: **end for**
-

3.6 Inference and Scores

At test time, the assistant and weight head are disabled; only T and S are used. The pixel-level and image-level discrepancies are

$$DS_1 = \frac{1}{C_4} \|F_T^4(\mathbf{x})_{i,j} - F_S^4(\mathbf{x})_{i,j}\|_2^2, \quad (10)$$

$$DS_2 = \frac{1}{C_4WH} \sum_{i,j} \|F_T^4(\mathbf{x})_{i,j} - F_S^4(\mathbf{x})_{i,j}\|_2^2. \quad (11)$$

For assistant–teacher discrepancy (used during assistant training),

$$DS_3 = \frac{1}{C_4} \|F_T^4(\mathbf{x})_{i,j} - F_A^4(\mathbf{x})_{i,j}\|_2^2, \quad (12)$$

$$DS_4 = \frac{1}{C_4WH} \sum_{i,j} \|F_T^4(\mathbf{x})_{i,j} - F_A^4(\mathbf{x})_{i,j}\|_2^2. \quad (13)$$

To mitigate background noise, we follow [9]: use 10% of training images as a validation set to compute percentile anchors (q_a, q_b) . Specifically, let θ_a and θ_b be the anomaly score values corresponding to the q_a -th and q_b -th percentiles of the validation distribution. At test time, the normalized score $\hat{S}$ for a raw score S is computed via:

$$\hat{S} = \text{clip} \left(\frac{S - \theta_a}{\theta_b - \theta_a}, 0, 1 \right). \quad (14)$$

We also optionally inject the teacher’s pixel score map as a calibration term when reporting segmentation: $\mathbf{M} = \eta \hat{\mathbf{M}}_T + (1 - \eta) \hat{\mathbf{M}}_{S-T}$ with $\eta \in [0, 1]$, where $\widehat{(\cdot)}$ denotes the aforementioned percentile normalization.

Complexity. Because the assistant and the dynamic weight head are training-only, TAD++ keeps the same inference complexity as a standard teacher–student pipeline.

4 Experiments

4.1 Datasets

We evaluate on 30 industrial scenarios from MVTec AD (15 classes; 5 textures and 10 objects; >5k images), VisA (12 objects; >10k images), and BTAD (3 classes; >2k images). Training uses only normal images; test sets contain both normal and defective images. All datasets provide image-level labels and pixel-level masks; we thus report image-level detection and pixel-level segmentation. Because textures and objects behave differently, we

additionally report averages on MVTec AD textures vs. objects.

4.2 Experimental Details

All images are normalized by ImageNet mean/std and resized to 768×768 . Following the pipeline in Sec. 3, one class is trained at a time. Features are extracted from the output of the final MB-Conv block at Stage 8 (stride 32) of EfficientNet-B5 to obtain a $304 \times 24 \times 24$ map, concatenated with a sinusoidal positional encoding of size $32 \times 24 \times 24$.

We use Adam with batch size 8. Teacher and assistant are trained for 240 epochs each; student and weight head are trained alternately for a total of 720 epochs, with half for S and half for the weight head. Learning rates for T , A , and S are set to 2×10^{-4} with weight decay 10^{-5} ; the rate is decayed to $0.1 \times$ in the last 10% epochs. The weight head uses learning rate 10^{-3} and weight decay 10^{-4} . Percentile normalization uses $q_a=0.9$ and $q_b=0.995$, and the hard-pixel quantile is $q_{\text{hard}}=0.7$.

4.3 Evaluation Metrics

We adopt AUROC. With $\text{FPR} \in [0, 1]$, the area is

$$\text{AUROC} = \int_0^1 \text{TPR}(u) du, \quad u = \text{FPR}. \quad (15)$$

Image-level AUROC is computed from DS_2 and pixel-level AUROC from DS_1 (Eqs. (10) and (11)).

4.4 Results

To ensure a strictly fair comparison, we re-implemented and re-evaluated the primary baseline, AST [10], along with leading state-of-the-art methods EfficientAD [9] and PatchCore within our experimental environment. Crucially, all these key baselines were tested using the exact same input resolution (768×768), data preprocessing pipeline, and hardware configuration (RTX 3090) as our TAD++ method. This eliminates potential discrepancies arising from different implementation details or resolution settings in the original papers. Consequently, the results reported for TAD++, AST, EfficientAD, and PatchCore in Tables 1–3 are obtained from our controlled reproduction and are presented as the average over five independent runs. For other older methods where re-evaluation was not feasible,

Table 1. Comparison on MVTec ad (image- / pixel-level auROC). best in bold.

Method	MKD	CutPaste	PyramidFlow (FNF)	PUAD (M)	D3AD	AST	OURS
Grid	78.01/91.78	100.00 /97.50	—/95.70	100.00 /—	100.00 / 99.20	98.21/96.92	99.36/99.07
Leather	95.05/98.05	100.00 /99.50	—/ 99.60	100.00 /—	98.50/99.40	100.00 /97.56	100.00 /98.28
Tile	91.57/82.77	94.60/90.50	—/ 97.90	99.86/—	95.50/94.70	99.71/92.37	99.99 /93.59
Carpet	79.25/95.64	93.90/ 98.30	—/90.80	95.22/—	94.20/97.60	97.15/96.51	99.40 /98.17
Wood	94.29/84.80	99.10/95.50	—/93.80	99.12/—	99.70/ 95.90	99.43/85.16	99.85 /92.10
Text. Avg.	87.63/90.61	97.52/96.26	—/95.56	98.84/—	97.58/ 97.36	98.90/93.70	99.72 /96.24
Bottle	99.39/96.32	98.20/97.60	—/95.90	99.92/—	100.00 / 98.60	100.00 /91.40	100.00 /94.27
Capsule	80.46/95.86	98.20/97.40	—/96.10	98.05/—	96.60/97.90	99.36/98.03	99.56 / 98.83
Pill	82.70/89.63	94.90/95.70	—/96.20	98.01/—	99.20 / 98.20	98.98/95.36	98.40/97.46
Transistor	85.55/76.45	96.10/93.00	—/ 97.40	99.67 /—	96.80/95.60	98.84/95.25	98.99/88.99
Zipper	93.24/93.90	99.90 / 99.30	—/95.40	98.87/—	98.20/98.30	99.20/96.07	99.78/98.14
Cable	89.19/82.40	81.20/90.00	—/92.10	97.77/—	97.80/93.30	98.46/94.89	98.83 / 96.69
Hazelnut	98.37/94.62	98.30/97.30	—/98.00	99.57/—	98.00/ 98.80	99.82/97.20	100.00 /98.40
Metal Nut	73.58/86.38	99.90 /93.10	—/ 97.20	98.48/—	98.90/96.10	98.98/91.26	99.71/96.32
Screw	83.31/95.96	98.70/96.70	—/96.60	92.64/—	83.90/99.00	99.26 /97.66	99.11/ 99.55
Toothbrush	92.17/96.12	99.40/98.10	—/98.90	100.00 /—	100.00 / 99.00	96.07/97.90	97.14/98.49
Obj. Avg.	87.80/90.76	95.48/95.82	—/96.18	98.30/—	96.94/ 97.48	98.89/95.50	99.15 /96.71
All Avg.	87.74/90.71	96.16/95.97	—/95.97	98.48/—	97.15/ 97.44	98.89/94.90	99.35 /96.56

results are cited directly from their respective original publications.

On averages, our method performs strongly on both textures and objects. Relative to AST, we improve most classes in terms of detection accuracy. However, we observe that the pixel-level segmentation AUROC on the Transistor category is lower than the baseline. We attribute this specific performance drop to the regularization effect of the inverse-distillation branch (Path B). While the reconstruction loss (L_{rec}) is highly effective at suppressing anomaly leakage in continuous textures, its objective tends to smooth out high-frequency details. Transistor defects often manifest as extremely fine, wire-thin anomalies; the student network, driven by L_{rec} , may aggressively ‘repair’ these thin structural deviations into the background, leading to false negatives. This reveals a trade-off between robust texture learning and fine-grained edge preservation.

Figure 6 visualizes several categories, where our method localizes defects accurately. Failure cases in Figure 7 reveal two main limitations. First, ‘shape clutter’ creates high-frequency gradients that the flow-based teacher struggles to map effectively to a standard normal distribution, occasionally resulting in false positives in non-defective

complex regions. Second, the inverse-distillation branch, while highly effective for continuous textures, can over-regularize faint anomalies, such as subtle scratches on the Capsule. In these instances, the reconstruction objective may treat the faint defect as high-frequency noise to be suppressed rather than a structural anomaly to be flagged, leading to false negatives. We leave the resolution of these specific corner cases to future work focusing on frequency-selective reconstruction.

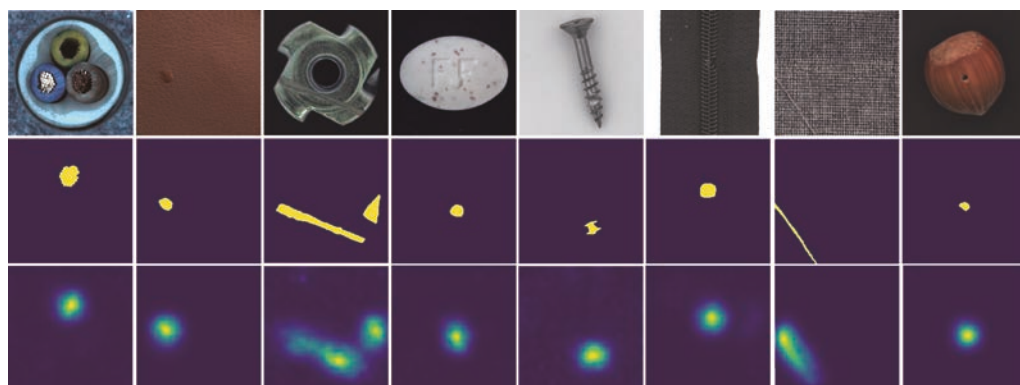
To explicitly validate our observations regarding these environmental variations, Table 4 quantifies the detection performance across subsets of categories specifically characterized by specular highlights, background clutter, and thin structural defects. As shown, our method achieves consistent gains in scenarios with high-frequency clutter and specular reflections, validating the efficacy of the heterogeneous assistant. However, we explicitly acknowledge a performance trade-off in the ‘Thin defects’ subset (e.g., Transistor). This quantitative drop aligns with our analysis above, confirming that the inverse-distillation branch’s reconstruction objective tends to over-smooth extremely fine anomalies.

Table 2. Comparison on btad (image- / pixel-level auROC). Best in bold.

Method	PyramidFlow (FNF)	PNI	D3AD	RealNet (DTD)	AST	OURS
1	100.00 /97.40	—/97.40	98.90/—	100.00 / 98.10	99.42/91.81	99.44/96.98
2	88.20/ 97.60	—/97.00	87.00/—	87.50/96.30	73.43/95.41	89.11 /97.39
3	99.30/98.10	—/99.00	99.70/—	99.40/99.60	99.19/82.66	99.81 / 99.69
All Avg.	95.80/97.70	—/97.80	95.20/—	95.70/98.00	90.68/89.96	96.12 / 98.02

Table 3. Comparison on Visa (image- / pixel-level auROC). best in bold.

Method	FAIRnoDTD	SuperSimpleNet	D3AD	RealNet (DTD)	AST	OURS
Candle	—/98.00	97.10/—	95.60/—	95.00/99.00	94.68/94.15	98.63 / 99.41
Capsules	—/93.90	81.50/—	88.50/—	88.10/97.60	87.75/96.11	91.23 / 99.41
Cashew	—/98.80	93.00/—	94.20/—	95.90/97.60	96.86 /97.96	96.38/ 98.99
Chewing gum	—/98.30	99.30/—	99.70/—	100.00 / 99.80	100.00 /98.49	99.96/99.01
Fryum	—/ 96.80	96.80/—	96.50/—	95.30/95.20	93.52/86.07	98.36 /96.54
Macaronil	—/99.70	93.10/—	94.30/—	98.20 /99.70	88.55/94.39	97.23/ 99.74
Macaroni2	—/91.70	75.00/—	92.50 /—	91.80/99.30	72.20/82.88	84.60/ 99.68
Pcb1	—/ 99.70	96.90/—	97.70/—	97.10/99.40	98.87/99.30	98.85/ 99.70
Pcb2	—/ 98.90	97.50/—	98.30 /—	97.50/97.80	97.11/96.33	97.08/98.81
Pcb3	—/ 99.30	94.40/—	97.40/—	97.60 /98.40	84.93/91.55	97.53/99.14
Pcb4	—/97.80	98.40/—	99.80 /—	99.20/98.60	99.67/93.68	99.48/ 98.67
Pipe fryum	—/ 99.40	97.60/—	96.90/—	99.90/98.60	99.96 /98.02	98.90/98.83
All Avg.	—/97.69	93.38/—	95.95/97.90	96.30/98.40	92.84/94.08	96.52 / 98.99

**Figure 6.** Visualization examples. Col. 1: input; Col. 2: ground truth; Col. 3: predictions from TAD++.

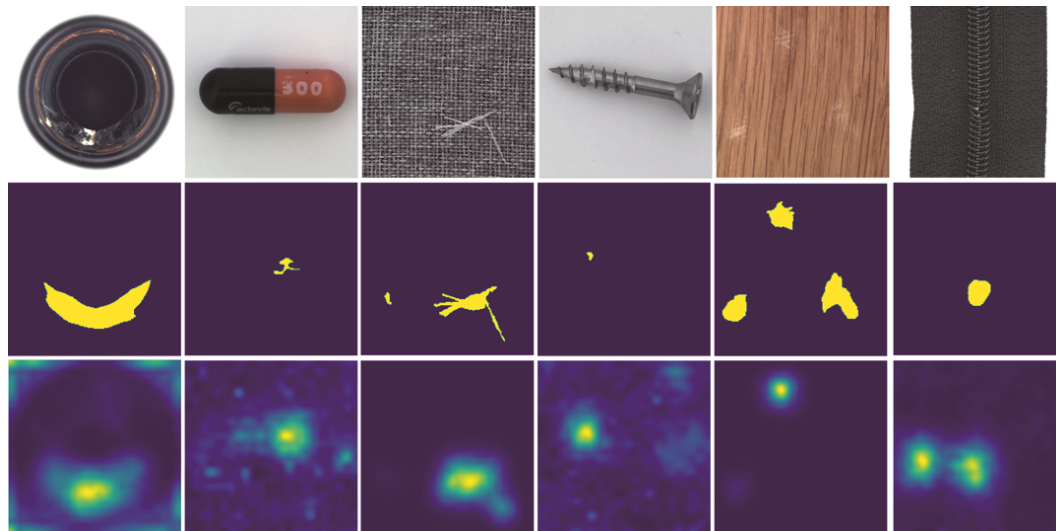


Figure 7. Failure cases. Col. 1: input; Col. 2: ground truth; Col. 3: predictions from TAD++.

Table 4. Quantitative robustness under nuisance factors on mvtec ad. We report auroc (%) on selected subsets. δ denotes tad++ minus ast. Best in bold.

Nuisance Factor	Subset (Categories)	Image-level AUROC (%)			Pixel-level AUROC (%)		
		AST	TAD++	Δ	AST	TAD++	Δ
Specular highlights	<i>e.g.</i> Bottle, Capsule, Screw, Pill, Metal Nut	98.76	99.36	+0.60	94.72	97.29	+2.57
Background clutter	<i>e.g.</i> Carpet, Grid, Leather, Tile, Wood	98.90	99.72	+0.82	93.70	96.24	+2.54
Thin defects	<i>e.g.</i> Transistor, Grid, Zipper	98.75	99.38	+0.63	96.08	95.40	-0.68
All MVTec AD	Average (15 classes)	98.89	99.35	+0.46	94.90	96.56	+1.66

Note: Values are aggregated from Table 1. While TAD++ excels in specular and cluttered scenarios, pixel-level performance on thin defects shows a trade-off due to inverse-distillation regularization.

4.5 Ablation Studies

4.5.1 Quantile Factor and Its Target

We study the hard-pixel quantile q_{hard} and where it is applied (student vs. weight head). Results in Table 5 show that applying q_{hard} only to the weight head works best; $q_{\text{hard}}=0.7$ is optimal across datasets.

Table 5. Image-level auroc on mvtec ad, btad, and visa while varying the target of q_{hard} . wn: weight network; s: student. best in bold.

Dataset	Target	0.9	0.8	0.7
MVTec AD	WN & S	98.77	98.98	99.00
	only WN	99.15	99.05	99.35
BTAD	WN & S	95.42	95.78	95.58
	only WN	96.02	95.56	96.12
VisA	WN & S	95.60	96.15	96.26
	only WN	96.49	96.46	96.52

4.5.2 Network Architecture

We ablate heterogeneous vs. homogeneous choices for assistant/student. Table 6 shows that a large-capacity student absorbs distilled knowledge better, and heterogeneity is beneficial.

Table 6. Assistant/student selection (image-level auroc). T: teacher; a: assistant structure; s: student structure. Best in bold.

Network	T-A-A	T-S-S	T-S-A	OURS (T-A-S)
MVTec AD	98.06	98.89	98.22	99.35
BTAD	95.51	95.24	95.57	96.12
VisA	93.69	95.41	94.08	96.52
All Avg.	95.75	96.51	95.96	97.33

4.5.3 Single Teacher vs. TAS

Table 7 shows that teacher–student already boosts the teacher, and adding the assistant yields further gains.

Table 7. Teacher only vs. T-S VS. T-A-S (image-level auroc). Best in bold.

Network	T	T-A	T-S	OURS (T-A-S)
MVTec AD	89.43	98.29	97.35	99.35
BTAD	95.13	95.21	95.29	96.12
VisA	87.17	94.06	95.15	96.52
All Avg.	90.58	95.85	95.93	97.33

4.5.4 Assistant Distillation Blocks

We vary which teacher blocks supervise the assistant, as detailed in Table 8. Distilling only the last block works best overall, avoiding redundancy and directly guiding the student’s final output.

Table 8. Assistant training: which teacher blocks to distill (image-level auroc). Best in bold.

Block	1	2	3	All
MVTec AD	98.31	98.47	99.35	98.94
BTAD	95.19	95.30	96.12	95.64
VisA	95.71	96.47	96.52	96.47
All Avg.	96.40	96.75	97.33	97.02

4.5.5 Dynamic Attention Weight Network

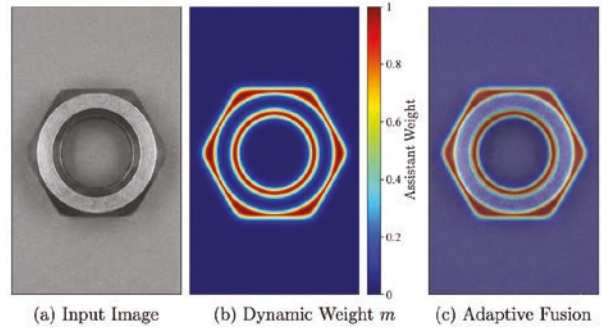


Figure 8. Visualization of the Dynamic Attention Weight Mechanism. To validate the adaptive behavior, we visualize the learned attention map on a sample structure. (a) Input image with distinct structural boundaries. (b) The learned attention map m , where red regions indicate higher weights assigned to the Assistant branch. (c) The adaptive fusion overlay demonstrates that the network prioritizes Assistant guidance along edges to compensate for the Teacher’s limitations.

Table 9 confirms that dynamic weighting outperforms fixed equal weights. To further investigate the internal mechanism, we provide an illustrative visualization in Figure 8. As observed, the network adaptively assigns higher weights (red regions) to structural boundaries where the flow-based teacher typically struggles, while assigning lower weights to uniform backgrounds. This confirms that the module effectively fuses heterogeneous signals based on feature complexity.

Table 9. Effect of the weight network (image-level auroc). Best in bold.

Network	w/o WN	w/ WN
MVTec AD	99.24	99.35
BTAD	95.60	96.12
VisA	96.41	96.52
All Avg.	97.08	97.33

4.5.6 Teacher Pixel Scores at Test Time

Adding the teacher’s pixel-level scores as a calibration term improves segmentation AUROC in Table 10.

Table 10. Adding teacher pixel scores during testing (pixel-level auroc). Ts: teacher pixel score. Best in bold.

Network	w/o TS	w/ TS
MVTec AD	96.16	96.56
BTAD	97.35	98.02
VisA	96.29	98.99
All Avg.	96.60	97.85

4.5.7 Impact of the Inverse-Distillation Branch

To explicitly verify the contribution of the pseudo-defect inverse-distillation branch (Path B), we conducted an isolate-and-measure study. In this experiment, we maintained the heterogeneous Teacher–Assistant–Student architecture and the Dynamic Attention Weight Network (Path A) but removed the inverse-distillation signal. Specifically, we set the balancing weight $\gamma = 0$ in Eq. (9), thereby eliminating the reconstruction loss L_{rec} during the student’s optimization.

The results, summarized in Table 11, demonstrate a consistent performance drop when the inverse branch is removed. Without the regularization provided by the pseudo-defect reconstruction task, the average image-level AUROC decreases from 97.33% to 96.63%. We observe the most significant drop on the BTAD dataset (−0.82%), suggesting that the inverse branch is particularly effective in scenarios with complex textures where standard distillation might suffer from anomaly leakage. These findings confirm that forcing the student to reconstruct normal features from defect-injected inputs serves as a crucial regularizer, anchoring the student to the normal manifold and enhancing detection sensitivity.

Furthermore, to address concerns regarding the sensitivity to pseudo-defect generation parameters, we evaluated TAD++ under ‘mismatched’ synthesis protocols to verify that improvements are not overfitting a specific generation strategy. We tested two suboptimal variants: (i) Large Scale Mismatch, where the pseudo-defect area ratio was drastically increased to $[0.1, 0.4]$ (simulating out-of-distribution sizes), and (ii) Coarse Synthesis, which employed raw cut-and-paste operations without boundary smoothing or erasing. Under these challenging settings, the average image-level AUROC decreased only marginally to 97.05% and 96.98%, respectively, compared to 97.33% with the default settings. Crucially, Table 11 shows that both variants still consistently outperform the baseline without the inverse branch at 96.63%. This confirms that the inverse-distillation branch functions as a robust regularizer: its benefit stems from the general task of restoring normality from corruption, rather than precise alignment between synthesized and real defect statistics.

Table 11. Ablation study isolating the effect of the inverse-distillation branch (path b). We report the image-level auroc (%) across all datasets. The “full tad++” row corresponds to our default setting with $\gamma > 0$.

Method Variant	MVTec AD	BTAD	VisA	All Avg.
T–A–S w/o Path B ($\gamma = 0$)	98.70	95.30	95.90	96.63
TAD++ (Full)	99.35	96.12	96.52	97.33
<i>Gain from Path B</i>	<i>+0.65</i>	<i>+0.82</i>	<i>+0.62</i>	<i>+0.70</i>

4.5.8 Sensitivity Analysis

To validate the robustness of our architectural choices, we conducted a sensitivity analysis on the input resolution and backbone capacity, as summarized in Table 12.

Comparing the default 768×768 setting with a lower resolution of 512×512 , we observe a slight decline in the average Image-level AUROC from 97.33% to 96.85%. This drop is primarily attributed to the loss of high-frequency details in categories with fine-grained defects, such as Transistor and Grid, confirming that higher resolution is critical for detecting subtle industrial anomalies. Conversely, increasing the resolution to 1024×1024 yields only marginal performance gains (+0.08%) but nearly doubles the training time (16.8h) and peak memory usage (38.4 GB). Thus, the 768×768 resolution represents the optimal trade-off between detection precision and computational feasibility.

Replacing the EfficientNet-B5 backbone with lighter alternatives, such as ResNet-50 or EfficientNet-B3, results in a performance decrease to 95.92% and 96.60%, respectively. However, it is crucial to note that even with the standard ResNet-50 backbone, TAD++ still demonstrates performance superior to the AST baseline’s average. This evidences that the performance improvements of TAD++ stem primarily from the proposed dual-path distillation and dynamic weighting mechanisms, rather than being solely dependent on the strong feature extraction capabilities of EfficientNet-B5.

4.6 Complexity and Efficiency Analysis

To address concerns regarding computational overhead and to demonstrate the practical viability of our method, we rigorously evaluate training costs in terms of time and memory, as well as inference efficiency in terms of latency and throughput. To ensure a strictly fair comparison, all methods—including the AST baseline and competing approaches like EfficientAD and PatchCore, were re-evaluated on the exact same hardware environment: a workstation equipped with an Intel Xeon Gold CPU and a single NVIDIA GeForce RTX 3090 GPU (24GB VRAM). All models were trained and tested with an input resolution of 768×768 and a batch size of 8.

As presented in Table 13, we explicitly break down the training costs. We acknowledge that TAD++ requires approximately $2.6\times$ the total training time of the AST baseline (8.4 hours vs. 3.2 hours). This increase stems from our specific three-phase optimization strategy: training the Teacher (Phase A), distilling the heterogeneous Assistant (Phase B), and the alternating optimization of the Student and Dynamic Weight Network (Phase C). Additionally, the peak VRAM usage during training reaches 21.2 GB, as the optimization of Phase C requires holding the Teacher, Assistant, and Student in memory simultaneously.

However, we argue that the increased offline training cost is a justifiable trade-off for industrial applications, where model updates are infrequent but online inference speed is critical. A key advantage of TAD++ is its zero-cost deployment. As illustrated in our framework design, the Assistant network, the Inverse Distillation branch, and the Dynamic Weight Network are strictly training-only modules. They are discarded after training is complete. Consequently, at test time, TAD++ utilizes only the Teacher and Student networks.

As shown in the rightmost columns of Table 13, TAD++ achieves identical inference parameters, VRAM usage, and throughput to the AST backbone. Compared to memory-bank-based methods like PatchCore (12.8 FPS at 768×768), our method offers significantly faster inference speeds suitable for real-time inspection. This confirms that TAD++ improves detection performance and robustness without imposing any additional computational burden during the deployment phase.

Conclusion

We presented TAD++, a dual-path distillation framework for unsupervised defect detection. Beyond a heterogeneous Teacher–Assistant–Student (T–A–S) pipeline, TAD++ incorporates a pseudo-defect inverse-distillation branch that acts as a regularizer to anchor the student to the normal manifold, and a dynamic attention weight network that adaptively resolves conflicts between teacher and assistant guidance.

Crucially, our design strategy prioritizes deployment efficiency. As demonstrated in our complexity analysis in Sec. 4.6, the assistant, the inverse

Table 12. Sensitivity analysis to input resolution and backbone selection. We report the average performance across mvtec ad, btad, and visa datasets. Results are evaluated under the same training recipe and hardware setting. The default configuration used in the main paper is marked with **Default**.

Setting variant	Backbone	Resolution	Image AUROC $\uparrow$	Pixel AUROC $\uparrow$	Train Time [h] $\downarrow$	Peak VRAM [GB] $\downarrow$
Default	EfficientNet-B5	768×768	97.33	97.85	8.4	21.2
<i>Sensitivity to Input Resolution (Backbone fixed to B5)</i>						
Low Res.	EfficientNet-B5	512×512	96.85	97.10	5.2	14.5
High Res.	EfficientNet-B5	1024×1024	97.41	97.92	16.8	38.4
<i>Sensitivity to Backbone Architecture (Resolution fixed to 768)</i>						
Lightweight	EfficientNet-B3	768×768	96.60	96.95	6.5	16.8
Standard	ResNet-50	768×768	95.92	96.40	5.8	15.2
Modern	ConvNeXt-T	768×768	96.45	96.80	6.9	17.5

Table 13. Complexity and efficiency analysis. We report the training time breakdown, peak memory usage, and inference metrics averaged across mvtec ad categories. Note: while tad++ incurs higher offline training costs due to its multi-phase heterogeneous design, it maintains identical inference efficiency (params, vram, fps) to the ast baseline, as the auxiliary assistant and weight network are removed during deployment.

Method	Training Phase (Cost)				Inference Phase (Efficiency)			
	Phase A [h]	Phase B [h]	Phase C [h]	Total [h]	Peak VRAM [GB]	Params [M]	VRAM [GB]	FPS
AST [8]	-	-	-	3.2	18.5	44.6	2.4	38.5
TAD++ (Ours)	1.8	1.2	5.4	8.4	21.2	44.6	2.4	38.5
EfficientAD [7]	-	-	-	6.5	12.0	12.5	1.1	95.2
PatchCore	-	-	-	0.5	14.2	30.4	4.5	12.8

branch, and the weight head are strictly training-only. While this multi-phase optimization increases offline training time compared to standard T-S baselines, it incurs zero additional inference latency and memory overhead, making TAD++ highly suitable for real-time industrial inspection. Extensive experiments on MVTec AD, BTAD, and Visa validate this trade-off, reporting state-of-the-art or highly competitive results.

We also frankly acknowledge the limitations. While the inverse-distillation branch effectively reduces anomaly leakage, its regularization effect may lead to over-smoothing on fine-grained high-frequency edges. Additionally, the increased training budget requires sufficient offline computational resources. Future work will focus on addressing these issues by exploring boundary-aware refinement to preserve edge details and investigating lighter backbones to reduce the training overhead without compromising the established inference ef-

ficiency.

Acknowledgments

This work was supported by Provincial Key Laboratory Performance Subsidy Project under grant 22567612H.

Declarations

- Funding: This study received funding from the Provincial Key Laboratory Performance Subsidy Project under grant 22567612H.
- Conflict of interest/Competing interests: The authors have no competing interests to declare that are relevant to the content of this article.
- Ethics approval: Not applicable.
- Consent to participate: Not applicable.
- Consent for publication: Not applicable.
- Availability of data and materials: The data and materials supporting this study will be publicly

released upon acceptance via a permanent open repository, and the corresponding access link will be provided in the final version.

- Code availability: The source code and pre-trained models will be made publicly available upon acceptance of this paper.
- Authors' contributions: Zhan Xin designed the research experiments and conducted the paper writing. The other authors provided guidance and supervision.

References

- [1] J. Ghanim, M. Awad, An unsupervised anomaly detection in electricity consumption using reinforcement learning and time series forest based framework, *Journal of Artificial Intelligence and Soft Computing Research* 15 (2025).
- [2] J. Zhang, H. Dong, J. Gao, R. Yao, G. Li, H. Wu, Self-organized operational neural networks for the detection of atrial fibrillation, *Journal of Artificial Intelligence and Soft Computing Research* 14 (2024).
- [3] W. Zhou, J. Hong, W. Yan, Q. Jiang, Modal evaluation network via knowledge distillation for no-service rail surface defect detection, *IEEE Transactions on Circuits and Systems for Video Technology* 34 (2024) 3930–3942.
- [4] Q. Yu, K. Zhu, Y. Cao, F. Xia, Y. Kang, T²: Few-shot text-free training-free defect image generation for industrial anomaly inspection, *IEEE Transactions on Circuits and Systems for Video Technology* 34 (2024) 11825–11837.
- [5] Y. Yang, D. Liu, mkdnad: A network flow anomaly detection method based on multi-teacher knowledge distillation, in: *2022 16th IEEE International Conference on Signal Processing (ICSP)*, volume 1, IEEE, 2022, pp. 314–319.
- [6] P. Bergmann, M. Fauser, D. Sattlegger, C. Steger, Uninformed students: Student-teacher anomaly detection with discriminative latent embeddings, in: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 4183–4192.
- [7] Y. Liu, W. Zhang, J. Wang, Adaptive multi-teacher multi-level knowledge distillation, *Neurocomputing* 415 (2020) 106–113.
- [8] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, Imagenet: A large-scale hierarchical image database, in: *2009 IEEE conference on computer vision and pattern recognition*, Ieee, 2009, pp. 248–255.
- [9] K. Batzner, L. Heckler, R. König, Efficientad: Accurate visual anomaly detection at millisecond-level latencies, in: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2024, pp. 128–138.
- [10] M. Rudolph, T. Wehrbein, B. Rosenhahn, B. Wandt, Asymmetric student-teacher networks for industrial anomaly detection, in: *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, 2023, pp. 2592–2602.
- [11] P. Bergmann, M. Fauser, D. Sattlegger, C. Steger, Mvtec ad—a comprehensive real-world dataset for unsupervised anomaly detection, in: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 9592–9600.
- [12] Y. Zou, J. Jeong, L. Pemula, D. Zhang, O. Dabeer, Spot-the-difference self-supervised pre-training for anomaly detection and segmentation, in: *European Conference on Computer Vision*, Springer, 2022, pp. 392–408.
- [13] P. Mishra, R. Verk, D. Fornasier, C. Piciarelli, G. L. Foresti, Vt-adl: A vision transformer network for image anomaly detection and localization, in: *2021 IEEE 30th International Symposium on Industrial Electronics (ISIE)*, IEEE, 2021, pp. 01–06.
- [14] M. Ke, C. Lin, Q. Huang, Anomaly detection of logo images in the mobile phone using convolutional autoencoder, in: *2017 4th International Conference on Systems and Informatics (ICSAI)*, IEEE, 2017, pp. 1163–1168.
- [15] Y. Li, W. Zhao, J. Pan, Deformable patterned fabric defect detection with fisher criterion-based deep learning, *IEEE Transactions on Automation Science and Engineering* 14 (2016) 1256–1264.
- [16] D. Abati, A. Porrello, S. Calderara, R. Cucchiara, Latent space autoregression for novelty detection, in: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 481–490.
- [17] Y. Qiu, T. Misu, C. Busso, Unsupervised scalable multimodal driving anomaly detection, *IEEE Transactions on Intelligent Vehicles* 8 (2022) 3154–3165.
- [18] T. Schlegl, P. Seeböck, S. M. Waldstein, U. Schmidt-Erfurth, G. Langs, Unsupervised anomaly detection with generative adversarial networks to guide marker discovery, in: *International conference on information processing in medical imaging*, Springer, 2017, pp. 146–157.

- [19] H. Chung, J. Park, J. Keum, H. Ki, S. Kang, Unsupervised anomaly detection using style distillation, *IEEE Access* 8 (2020) 221494–221502.
- [20] X. Wang, Y. Du, S. Lin, P. Cui, Y. Shen, Y. Yang, advae: A self-adversarial variational autoencoder with gaussian anomaly prior knowledge for anomaly detection, *Knowledge-Based Systems* 190 (2020) 105187.
- [21] D. Gong, L. Liu, V. Le, B. Saha, M. R. Mansour, S. Venkatesh, A. v. d. Hengel, Memorizing normality to detect anomaly: Memory-augmented deep autoencoder for unsupervised anomaly detection, in: *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 1705–1714.
- [22] W. Li, Z. Shang, J. Zhang, M. Gao, S. Qian, A novel unsupervised anomaly detection method for rotating machinery based on memory augmented temporal convolutional autoencoder, *Engineering Applications of Artificial Intelligence* 123 (2023) 106312.
- [23] K. Wu, L. Zhu, W. Shi, W. Wang, J. Wu, Self-attention memory-augmented wavelet-cnn for anomaly detection, *IEEE Transactions on Circuits and Systems for Video Technology* 33 (2023) 1374–1385.
- [24] M. Salehi, A. Eftekhari, N. Sadjadi, M. H. Rohban, H. R. Rabiee, Puzzle-ae: Novelty detection in images through solving puzzles, *arXiv preprint arXiv:2008.12959* (2020).
- [25] M. Haselmann, D. P. Gruber, P. Tabatabai, Anomaly detection using deep learning based image completion, in: *2018 17th IEEE international conference on machine learning and applications (ICMLA)*, IEEE, 2018, pp. 1237–1242.
- [26] C.-L. Li, K. Sohn, J. Yoon, T. Pfister, Cutpaste: Self-supervised learning for anomaly detection and localization, in: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 9664–9674.
- [27] M. Pei, N. Liu, B. Zhao, H. Sun, Self-supervised learning for industrial image anomaly detection by simulating anomalous samples, *International Journal of Computational Intelligence Systems* 16 (2023) 152.
- [28] Q. Ma, E. Zhang, Y. Chen, J. Duan, L. Shao, Sia-net: Structural information awareness network based on normal samples for surface defect detection, *Engineering Applications of Artificial Intelligence* 126 (2023) 107131.
- [29] Q. Chen, H. Luo, H. Gao, C. Lv, Z. Zhang, Progressive boundary guided anomaly synthesis for industrial anomaly detection, *IEEE Transactions on Circuits and Systems for Video Technology* (2024) 1–1.
- [30] B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, R. C. Williamson, Estimating the support of a high-dimensional distribution, *Neural computation* 13 (2001) 1443–1471.
- [31] D. M. Tax, R. P. Duin, Support vector data description, *Machine learning* 54 (2004) 45–66.
- [32] L. Ruff, R. Vandermeulen, N. Goernitz, L. Deecke, S. A. Siddiqui, A. Binder, E. Müller, M. Kloft, Deep one-class classification, in: *International conference on machine learning*, PMLR, 2018, pp. 4393–4402.
- [33] P. Perera, V. M. Patel, Learning deep features for one-class classification, *IEEE Transactions on Image Processing* 28 (2019) 5450–5463.
- [34] W. Mao, G. Wang, L. Kou, X. Liang, Deep domain-adversarial anomaly detection with one-class transfer learning, *IEEE/CAA Journal of Automatica Sinica* 10 (2023) 524–546.
- [35] J. Wang, X. Wang, R. Hao, H. Yin, B. Huang, X. Xu, J. Liu, Incremental template neighborhood matching for 3d anomaly detection, *Neurocomputing* 581 (2024) 127483.
- [36] Z. Hu, Z. Yin, L. Qin, F. Xu, A novel method of fault diagnosis for injection molding systems based on improved vgg16 and machine vision, *Sustainability* 14 (2022) 14280.
- [37] W. Ma, S. Lan, W. Huang, W. Wang, H. Yang, Y. Ma, Y. Ma, A semantics-aware normalizing flow model for anomaly detection, in: *2023 IEEE International Conference on Multimedia and Expo (ICME)*, IEEE, 2023, pp. 2207–2212.
- [38] M. Rudolph, B. Wandt, B. Rosenhahn, Same same but different: Semi-supervised defect detection with normalizing flows, in: *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, 2021, pp. 1907–1916.
- [39] M. Salehi, N. Sadjadi, S. Baselizadeh, M. H. Rohban, H. R. Rabiee, Multiresolution knowledge distillation for anomaly detection, in: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 14902–14912.
- [40] X. Zhang, S. Li, X. Li, P. Huang, J. Shan, T. Chen, Destseg: Segmentation guided denoising student-teacher for anomaly detection, in: *Proceedings*

- of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 3914–3923.
- [41] S. Yamada, S. Kamiya, K. Hotta, Reconstructed student-teacher and discriminative networks for anomaly detection, in: 2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, 2022, pp. 2725–2732.
- [42] T. Fukuda, M. Suzuki, G. Kurata, S. Thomas, J. Cui, B. Ramabhadran, Efficient knowledge distillation from an ensemble of teachers., in: Interspeech, 2017, pp. 3697–3701.
- [43] S. You, C. Xu, C. Xu, D. Tao, Learning from multiple teacher networks, in: Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining, 2017, pp. 1285–1294.
- [44] M.-C. Wu, C.-T. Chiu, K.-H. Wu, Multi-teacher knowledge distillation for compressed video action recognition on deep neural networks, in: ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2019, pp. 2202–2206.
- [45] H. Zhang, D. Chen, C. Wang, Adaptive multi-teacher knowledge distillation with meta-learning, in: 2023 IEEE International Conference on Multi-media and Expo (ICME), IEEE, 2023, pp. 1943–1948.
- [46] F. Yuan, L. Shou, J. Pei, W. Lin, M. Gong, Y. Fu, D. Jiang, Reinforced multi-teacher selection for knowledge distillation, in: Proceedings of the AAAI Conference on Artificial Intelligence, volume 35, 2021, pp. 14284–14291.
- [47] Y. Ma, X. Jiang, N. Guan, W. Yi, Anomaly detection based on multi-teacher knowledge distillation, Journal of systems architecture 138 (2023) 102861.
- [48] M. Pei, N. Liu, P. Gao, H. Sun, Reverse knowledge distillation with two teachers for industrial defect detection, Applied Sciences 13 (2023) 3838.
- [49] J. Hao, K. Huang, C. Chen, J. Mao, Dual-student knowledge distillation for visual anomaly detection, Complex & Intelligent Systems (2024) 1–13.
- [50] Q. Zhou, S. He, H. Liu, T. Chen, J. Chen, Pull & push: Leveraging differential knowledge distillation for efficient unsupervised anomaly detection and localization, IEEE Transactions on Circuits and Systems for Video Technology 33 (2023) 2176–2189.
- [51] M. Tan, Q. Le, Efficientnet: Rethinking model scaling for convolutional neural networks, in: International conference on machine learning, PMLR, 2019, pp. 6105–6114.
- [52] A. Waswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. Gomez, L. Kaiser, I. Polosukhin, Attention is all you need, in: NIPS, 2017.



Xin Zhan is a Ph.D. candidate at Yanshan University, Qinhuangdao, China. His current research interests lie in the areas of pattern recognition and image processing.
<https://orcid.org/0009-0001-4909-5720>



Wenming Zhang is an associate professor in the institute electrical engineering from the Yanshan University, Qinhuangdao, China. His current research interests lie in the area of computer vision and Pattern Recognition.
<https://orcid.org/0000-0003-3802-3534>



Yaqian Li is an associate professor in the institute electrical engineering from the Yanshan University, Qinhuangdao, China. Her current research interests lie in the area of Pattern Recognition.
<https://orcid.org/0000-0003-1032-9910>



Haibin Li is a professor in the institute electrical engineering from the Yanshan University, Qinhuangdao, China. His current research interests lie in the area of Pattern Recognition.
<https://orcid.org/0000-0002-0310-2892>



Ruihao Chen is a master's student at Yanshan University, Qinhuangdao, China. His current research interests lie in the area of computer vision.
<https://orcid.org/0009-0004-4838-9710>