DeCoFlow: Structural Decomposition of Normalizing Flows for Continual Anomaly Detection*

Hun Im[1], Jungi Lee[1], Subeen Cha[1], and Pilsung Kang[1]

Seoul National University, Seoul, Korea
{hun_im,jungi_lee,subeen_cha,pilsung_kang}@snu.ac.kr

Abstract. In industrial environments, new product categories arrive sequentially, requiring continual anomaly detection without access to past data. Normalizing Flows (NFs) provide exact density estimation but suffer from catastrophic forgetting as parameter updates across tasks distort the density manifold. While parameter isolation can prevent interference, it must preserve the strict invertibility and Jacobian validity of NFs. To satisfy these requirements, we exploit the inherent property that affine coupling layers maintain transformation validity regardless of subnet parameterization. Based on this, we propose DeCoFlow, which decomposes subnets into a frozen universal base and task-specific low-rank adapters to isolate updates. We further introduce Task-Specific Alignment, Auxiliary Coupling Layers, and Tail-Aware Loss to compensate for frozen-base rigidity. DeCoFlow achieves state-of-the-art image-level AUROCs of 98.40% on MVTec-AD and 93.00% on VisA, while maintaining parameter-level zero forgetting (0.00% FM under correct routing) with only 2.27M parameters per task.

Keywords: Continual Anomaly Detection · Anomaly Detection · Continual Learning · Normalizing Flows · Parameter-Efficient Fine-Tuning · Parameter Isolation

1 Introduction

In industrial manufacturing, product lines evolve over time, requiring anomaly detectors to accommodate new classes sequentially. Recent anomaly detection has moved from one-class to multi-class settings [36, 38], yet most methods assume a static data distribution, whereas practical manufacturing requires learning tasks sequentially under storage and privacy constraints [4], forcing models to learn without past data and leading to catastrophic forgetting [7, 23].

Normalizing Flows (NFs) have emerged as a powerful baseline because they provide exact density estimation and naturally avoid the identity mapping problem of reconstruction-based methods. However, extending NFs to continual learning is challenging as catastrophic forgetting is significantly more pronounced in

* Code is available at https://github.com/crimama/DeCoFlow.

density-estimation models. Unlike classification models that only need to preserve decision boundaries, NFs rely on maintaining the entire density manifold to score anomalies; even minor parameter interference during sequential updates can thus severely distort the learned density [14].

Because most existing CL methods [11, 16] were designed for discriminative tasks, applying them to NFs reveals a fundamental dilemma. Replay methods mitigate forgetting but introduce fidelity limitations that degrade the precise density estimation required by NFs [30]. Parameter-isolation methods prevent interference but lead to linear model growth [28]. Although parameter-efficient fine-tuning (PEFT) techniques like LoRA offer a scalable alternative [17, 26], applying them directly to NFs is constrained by the strict invertibility and exact Jacobian requirements of flow models [6].

To address this, we exploit a structural property of coupling-based NFs: transformation validity is preserved regardless of subnet parameterization. Based on this insight, we propose DeCoFlow, a framework that decomposes each scale/shift subnet into a frozen universal base and learnable task-specific low-rank residuals [6]. This subnet-level decomposition preserves invertibility and Jacobian validity (Sec. 3) while enabling task-wise adapter isolation. To compensate for frozen-base rigidity, we introduce Task-Specific Alignment (TSA), Auxiliary Coupling Layers (ACL), and Tail-Aware Loss (TAL) (Sec. 4). Our main contributions are:

- **Structural Design Principle:** We establish that subnet independence in coupling-based NFs enables parameter isolation for continual learning, empirically confirming this architectural specificity through cross-architecture comparisons.
- **DeCoFlow Framework:** We propose DeCoFlow, decomposing each coupling subnet into a frozen universal base and task-specific low-rank adapters, achieving parameter-level zero forgetting under correct routing with 2.5% (2.27M) overhead per task.
- **Frozen-Regime Compensators:** To mitigate frozen-base rigidity, we introduce TSA for input alignment, ACL for residual correction, and TAL for capacity redistribution, validating their synergy through factorial and block-wise analysis.

2 Related Work

2.1 Deep Anomaly Detection

Recent anomaly detection has transitioned to the Unified AD paradigm, covering multiple classes with a single model [36, 38]. However, existing reconstruction-based methods are vulnerable to the identity mapping problem, where even anomalous data is reconstructed. In contrast, Normalizing Flows (NFs) structurally avoid this through direct likelihood optimization and have demonstrated strong performance [37, 39]. While recent methods introduce GMMs [35] or vector quantization [40] for multi-class modeling, their fixed capacity limits adapta-

tion to evolving data. Continual learning is thus essential for dynamic industrial environments.

2.2 Continual Learning

Continual learning methods for mitigating catastrophic forgetting [7, 23] fall into three categories: regularization (EWC [11], MAS [1]), replay (GEM [21], DGR [30]), and architecture-based isolation including neuron masking [22, 29] and PEFT such as LoRA [9]. Among these, LoRA-based methods have gained prominence for structurally preventing parameter interference. Recent extensions include MoE-based routing (GainLoRA [17], MINGLE [26]), geometric constraints (AnaCP [24], CaLoRA [33]), and dynamic subspace allocation (CoSO [3]). However, these techniques target discriminative decision boundaries, and cannot be directly applied to density-based models without risking likelihood manifold collapse.

2.3 Continual Learning in Anomaly Detection

Continual anomaly detection methods extend the aforementioned CL paradigms. **Replay**-based approaches mitigate forgetting by retaining past data or generating synthetic samples, utilizing techniques like incremental coresets [34] or diffusion replay [10]. However, their effectiveness is strictly bounded by memory budgets and replay fidelity. Meanwhile, **constraint-driven** methods [14, 25] stabilize model updates to reduce forgetting, but inevitably suffer from the inherent stability-plasticity dilemma when learning new tasks. **Dynamic architecture** methods mitigate interference through structural changes. Parameter-isolation designs [12, 28] scale linearly with the number of tasks, whereas prompt-based variants [19, 41] remain overly sensitive to prompt quality. None of these approaches exploits the inherent structural properties of NFs for safe parameter isolation.

3 Structural Basis for Parameter Isolation

The structural basis for parameter isolation in our framework lies in the architectural independence of affine coupling layers. We exploit this property to decompose the subnets within Normalizing Flows (NFs). This approach is rooted in the mathematical guarantee that the coupling mechanism preserves invertibility and exact Jacobian computation regardless of its internal subnet architecture [6].

The key rationale is that coupling layer invertibility is guaranteed by the *external structure* of input splitting and recombination. When input $\mathbf{z}$ is split into $(\mathbf{z}_1, \mathbf{z}_2)$, the forward transformation is:

$$\mathbf{y}_1 = \mathbf{z}_1, \quad \mathbf{y}_2 = \mathbf{z}_2 \odot \exp\big(s(\mathbf{z}_1)\big) + t(\mathbf{z}_1), \tag{1}$$

Table 1: Cross-architecture comparison under the same frozen base + LoRA strategy.

Architecture	Decomp. Level	I-AUC (%)	P-AP (%)	FM (%p)
NF (DeCoFlow)	**Coupling-level**	**98.47**	**58.57**	**0.00**
NF (Feature-level)	Feature-level	50.00	4.15	0.00
Autoencoder	Decoder LoRA	68.15	20.52	0.00
VAE	Decoder LoRA	65.25	18.14	0.04
Teacher-Student	Student LoRA	62.03	15.34	18.98

where $s, t : \mathbb{R}^{d/2} \to \mathbb{R}^{d/2}$ are the scale and shift subnets. Since $\mathbf{y}_1 = \mathbf{z}_1$, the inverse is derived without s^{-1} or t^{-1}:

$$\mathbf{z}_1 = \mathbf{y}_1, \quad \mathbf{z}_2 = (\mathbf{y}_2 - t(\mathbf{y}_1)) \oslash \exp\big(s(\mathbf{y}_1)\big). \tag{2}$$

The lower-triangular Jacobian structure yields $\log|\det \mathbf{J}_f(\mathbf{z})| = \sum_i s_i(\mathbf{z}_1)$, which depends solely on the outputs of the scale function s. This structural independence, ensuring both invertibility and exact likelihood computation regardless of the internal architecture, is formalized as follows:

Proposition 1 (Subnet-Independent Validity [6]). *For the affine coupling transformation $f : \mathbb{R}^d \to \mathbb{R}^d$ defined by Eq.* (1)*, for any differentiable functions $s, t : \mathbb{R}^{d/2} \to \mathbb{R}^{d/2}$:*

(i) f is always invertible regardless of the subnet architecture of s and t;
(ii) $\log|\det \mathbf{J}_f(\mathbf{z})| = \sum_i s_i(\mathbf{z}_1)$, where the log-det computation formula itself is independent of t and of the internal architecture of s.

Since flow validity depends only on subnet *outputs*, the internal weights can be restructured to accommodate continual learning. We exploit this by decomposing each subnet into a shared base and a task-specific low-rank adapter:

Corollary 1 (Parameter-Level Forgetting Prevention under Correct Routing). *Let each subnet be decomposed as $\psi(\mathbf{z}) = \psi_{\text{base}}(\mathbf{z}; \theta_{\text{base}}) + \delta\psi_\tau(\mathbf{z}; \Delta\theta_\tau)$ for $\psi \in \{s, t\}$, where θ_{base} is frozen after Task 0, and each task-specific adapter $\Delta\theta_\tau$ is stored independently. When adapter $\Delta\theta_{\tau'}$ is loaded at inference, the flow output is identical to the state at the completion of task τ' training.*

This decomposition eliminates parameter-induced forgetting by isolating task-specific updates.

To verify this property is unique to coupling-based NFs, we apply the same frozen+LoRA strategy to other anomaly detection architectures (Tab. 1). AE and VAE yield suboptimal 65–68% I-AUC because freezing disrupts their encoder-decoder correspondence, while the Teacher-Student architecture suffers from an 18.98% forgetting rate due to shared encoder updates. Even within NFs, feature-level adaptation causes manifold collapse (I-AUC 50.00%), as the frozen density mapping cannot accommodate the shifted distribution. Only coupling-level decomposition achieves FM=0.00% with competitive detection, confirming that

subnet independence makes coupling NFs inherently suited for density-based continual learning.

However, a permanently frozen base restricts representational flexibility, biasing the model toward initial task statistics. To overcome this rigidity, the following section introduces compensatory mechanisms within the DeCoFlow framework.

4 Proposed Method: DeCoFlow

4.1 Problem Formulation

Continual Anomaly Detection addresses T tasks, $\mathcal{D} = \{\mathcal{D}_0, \ldots, \mathcal{D}_{T-1}\}$ arriving sequentially. Each task $\mathcal{D}_\tau$ contains only normal samples and prior data $\mathcal{D}_{0:\tau-1}$ is inaccessible. We follow the Class-Incremental Learning (CIL) scenario, where Task IDs are unavailable at inference. The NF parameters are denoted as $\theta = \{\theta_{\text{base}}, \{\Delta\theta_\tau, \phi_\tau\}_{\tau=0}^{T-1}\}$, with $\Delta\theta_\tau$ as task-specific LoRA parameters and $\phi_\tau = \{\gamma_\tau, \beta_\tau, \theta_{\text{ACL},\tau}\}$ as auxiliary adaptation parameters (TSA and ACL). For each task $\tau \geq 1$, θ_{base} is frozen and only $\Delta\theta_\tau, \phi_\tau$ are optimized.

4.2 Framework Overview

DeCoFlow (Fig. 1) achieves robust continual anomaly detection by coupling structural isolation with compensatory modules. At its core, the Decomposed Coupling Layer (DCL) partitions subnets into a frozen base and task-specific adapters to eliminate parameter interference. To mitigate frozen-base rigidity, we integrate three components: Task-Specific Alignment (TSA) for input calibration, Auxiliary Coupling Layers (ACL) for residual density correction, and Tail-Aware Loss (TAL) for enhanced learning on challenging samples. During inference, a prototype-based routing mechanism identifies the appropriate adapter via Mahalanobis distance, enabling task-agnostic anomaly detection in a single forward pass.

4.3 Feature Extraction and Preprocessing

Multi-scale features from a pre-trained backbone are aggregated into unified patch representations $F \in \mathbb{R}^{|\mathcal{B}| \times H \times W \times D}$ [27]. Unlike previous methods employing separate NFs per scale [8,37], DeCoFlow processes these integrated features through a single NF, minimizing parameter overhead to one LoRA adapter per task. The backbone remains frozen, and we incorporate 2D sinusoidal positional encodings to preserve essential spatial context within the patch grid.

Task-Specific Alignment (TSA) The NF base weights, frozen after Task 0 (Sec. 3), remain biased toward initial task statistics, potentially making subsequent inputs appear out-of-distribution. TSA mitigates this shift by standardizing patch features via LayerNorm followed by a task-specific affine transformation $(\gamma_\tau, \beta_\tau)$:

$$f_{\text{LN}} = \frac{F - \mathbb{E}[F]}{\sqrt{\text{Var}[F] + \epsilon}}, \quad \hat{F}_\tau = \gamma_\tau \odot f_{\text{LN}} + \beta_\tau. \tag{3}$$

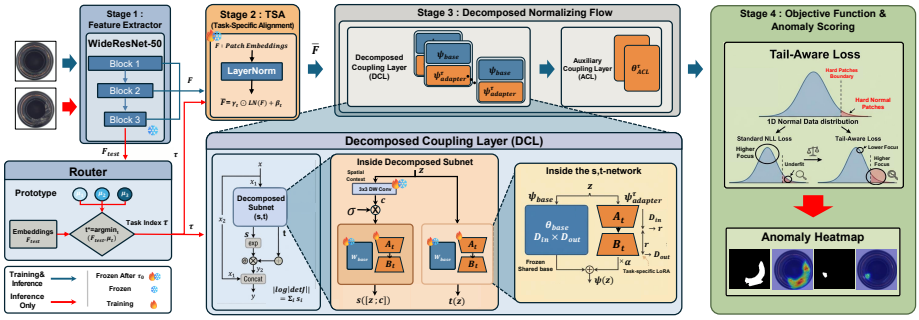

Fig. 1: Overall pipeline of DeCoFlow. Top: The sequential flow of multi-scale features through the frozen backbone, TSA calibration, and the Normalizing Flow stages. Middle: Detailed structural views of (A) the affine coupling process, (B) the scale subnet architecture with spatial context, and (C) the parallel $W_{\text{base}} + \Delta W$ decomposition in DCL. Bottom: The training timeline where the base is frozen after Task 0 to enable sequential adapter-only updates for subsequent tasks.

The parameters γ_τ and β_τ are constrained by sigmoid and tanh activations, respectively, to ensure stable and bounded outputs. Notably, TSA serves as an external preprocessing step rather than an inherent flow component. Because the NF treats $\hat{F}_\tau$ as a static input, LayerNorm statistics ($\mathbb{E}[F], \text{Var}[F]$) do not impact the log-determinant, thus preserving the coupling layers' original Jacobian structure.

4.4 Decomposed Coupling Layer

DCL serves as the core building block of DeCoFlow. By partitioning each coupling subnet into a frozen shared base and a task-specific adapter, it composes the subnets as the sum of these components, following the principles established in Sec. 3.

Internal Subnet Structure Each subnet $\psi \in \{s, t\}$ is decomposed into a frozen base and a task-specific adapter via an additive mapping:

$$\psi(\mathbf{z}) = \underbrace{\psi_{\text{base}}(\mathbf{z}; \theta_{\text{base}})}_{\text{shared across tasks}} + \underbrace{\delta\psi_\tau(\mathbf{z}; \Delta\theta_\tau)}_{\text{task-specific}}, \tag{4}$$

where the adapter $\delta\psi_\tau$ is implemented via LoRA [9] with zero-initialized $\mathbf{B}_\tau$. This ensures that each task initially inherits the frozen base distribution. Per Proposition 1, this decomposition preserves invertibility and Jacobian computation, as they remain invariant to the internal architecture of ψ.

Asymmetric Design with Context Injection While standard NFs typically employ symmetric architectures for both subnets, we adopt an asymmetric configuration to better align with their distinct roles in density-based anomaly detection. Specifically, the scale function (s) is responsible for localized defect capture through volume change, whereas the shift function (t) handles global

distribution translation. This functional distinction motivates the following differentiated structures:

$$s([\mathbf{z};\mathbf{c}]) = s_{\text{base}}([\mathbf{z};\mathbf{c}]) + \delta s_\tau([\mathbf{z};\mathbf{c}]). \tag{5}$$

Conversely, to ensure stable translation and noise robustness, t operates strictly on $\mathbf{z}$ without external context:

$$t(\mathbf{z}) = t_{\text{base}}(\mathbf{z}) + \delta t_\tau(\mathbf{z}). \tag{6}$$

To aggregate neighborhood information, $\mathbf{c}$ is computed by applying a 3×3 depthwise convolution on the spatially-reshaped 2D patch grid, followed by flattening the result back to 1D. The spatial context is further modulated by a learned gate $\eta = \eta_{\max}\,\sigma(\theta_c)$, which is frozen after Task 0 to ensure that the spatial augmentation remains consistent across all subsequent tasks.

Continual Learning Protocol In Task 0, θ_{base} and $\Delta\theta_0$ are jointly trained, after which the base is permanently frozen. For $\tau \geq 1$, only LoRA $\Delta\theta_\tau$ and TSA ϕ_τ are optimized, with $\mathbf{B}_\tau$ zero-initialized. This freezing strategy is justified by a high inter-task gradient cosine similarity of 0.78 (Tab. 6c), indicating that the learned base features are sufficiently task-invariant.

4.5 Auxiliary Coupling Layers (ACL)

Due to the structural constraint of affine coupling—where only half of the dimensions are transformed per step—complete cross-dimensional decorrelation is not inherently guaranteed. This is further compounded by the low-rank nature of LoRA, which restricts statistical adjustments to a rank-r subspace and may prevent full-rank correction across all dimensions. Consequently, residual correlations can accumulate through the DCL stack, a phenomenon empirically analyzed in Sec. 5.4 (Fig. 5b). To bridge this statistical gap, we introduce Auxiliary Coupling Layers (ACL) to provide additional residual density correction.

Auxiliary Coupling Layers (ACL) are a small number of affine coupling layers appended after the DCLs to compensate for these residual statistical discrepancies:

$$\mathbf{z}_{\text{out}} = g_{\text{ACL},\tau}(\mathbf{z}_{\text{DCL}};\, \theta_{\text{ACL},\tau}). \tag{7}$$

To bridge the remaining statistical gap, each task is assigned independent ACL parameters $\theta_{\text{ACL},\tau}$, as residual correlations are inherently task-specific. Each ACL subnet is implemented as a two-layer MLP with a hidden dimension $d_{\text{hidden}} = \lfloor D/2 \rfloor$. Crucially, the output layer is zero-initialized so that the module initially acts as an identity function. This allows ACL to introduce progressive statistical corrections without perturbing the density transformation already established by the DCL stack.

While adding ACL entails parameter overhead, it remains far more efficient than allocating independent networks for each task. Crucially, the statistical normalization in ACL is structurally complementary to DCL's nonlinear mapping, such that its role cannot be substituted by merely increasing the number of DCL blocks. Detailed analyses of this efficiency and modular synergy are provided in Sec. 5.4.

4.6 Training Objective: Tail-Aware Loss (TAL)

DeCoFlow is trained by minimizing the Negative Log-Likelihood (NLL) derived from the change-of-variables formula:

$$\mathcal{L}_{\text{NLL}} = -\log p_Z(\mathbf{z}) - \log|\det \mathbf{J}_f(\mathbf{x})| = \tfrac{1}{2}\|\mathbf{z}\|^2 - \log|\det \mathbf{J}_f(\mathbf{x})|, \tag{8}$$

where $\mathbf{z} = f(\mathbf{x})$ is the latent output and $p_Z = \mathcal{N}(\mathbf{0}, \mathbf{I})$ is the standard Gaussian prior (constant omitted). However, standard NLL optimization often prioritizes dominant high-likelihood modes, causing limited adapter capacity to be over-allocated to the distribution center. This results in underfitting low-density tail regions, which are critical for distinguishing normal samples from anomalies.

To mitigate this capacity imbalance, we introduce Tail-Aware Loss (TAL), drawing inspiration from hard-example mining techniques such as OHEM [31] and Focal Loss [18]. TAL reweights high-loss patches to redirect the constrained representational capacity of LoRA subnets toward under-represented tail distributions:

$$\mathcal{L}_{\text{TAL}} = (1-\omega)\,\mathbb{E}_{x\in\mathcal{B}}[\mathcal{L}_{\text{NLL}}(x)] + \omega\,\mathbb{E}_{x\in\mathcal{K}}[\mathcal{L}_{\text{NLL}}(x)], \tag{9}$$

where ω is the tail weight ratio, $\mathcal{B}$ is the full batch, and $\mathcal{K}$ denotes the set of top-k highest NLL-loss patches with $k = \lfloor |\mathcal{B}|HW \cdot r_{\text{tail}} \rfloor$. We set $\omega = 0.85$ for MVTec-AD, $\omega = 0.8$ for VisA, and $r_{\text{tail}} = 0.02$. These parameters remain fixed across all tasks as the tail ratio of normal distributions is empirically stable, while dynamic adjustment remains a subject for future work.

Since TAL amplifies gradients on tail patches, scale function outputs can lead to uncontrolled volume expansion. To ensure numerical stability and stable density estimation on the frozen base, we introduce an ℓ_2 regularization term on the log-Jacobian determinant. The final objective is defined as:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{TAL}} + \lambda_{\text{reg}}\,\|\log|\det \mathbf{J}_f(\mathbf{z})|\|_2^2\,, \tag{10}$$

where $\lambda_{\text{reg}} = 10^{-4}$ serves to constrain the Jacobian magnitude.

4.7 Task Routing and Inference

Prototype-Based Task Routing To identify the appropriate task-specific adapter without auxiliary networks, we employ prototype-based matching. For each task τ, a prototype is defined by the mean μ_τ and covariance Σ_τ of normal feature statistics computed from the training set. At inference, the routing mechanism selects the optimal task index τ^* by minimizing the Mahalanobis distance between the test feature f_{test} and the stored prototypes:

$$\tau^* = \arg\min_{\tau} \sqrt{(f_{\text{test}} - \mu_\tau)^T \Sigma_\tau^{-1} (f_{\text{test}} - \mu_\tau)}. \tag{11}$$

Specifically, we use only the final scale of the multi-scale features as the routing query f_{test} to capture the global task context. Although the subsequent NF process utilizes multi-scale features to preserve dense structural details, routing

relies solely on this high-level semantic representation. This strategy ensures accurate task identification within a single forward pass, effectively eliminating the requirement for an external classifier or additional backbone computations.

Since the score is computed through the routed adapter, correct routing is required to realize the zero-forgetting property: misrouting does degrade performance, as forcing 5% misrouting on MVTec-AD lowers P-AP from 56.93% to 30.20%. In practice, however, every class retains a sufficient routing margin—even the worst case (VisA *pcb2*) keeps its second-nearest prototype 2.58× farther than the nearest—so Mahalanobis routing reaches 100% accuracy on both datasets.

Anomaly Scoring Patch-level anomaly scores are defined as NLL $a_{h,w} = -\log p(z_{h,w}) - \log|\det \mathbf{J}_{h,w}|$. The image-level score aggregates the top-K patches such that $a_{\text{img}} = \frac{1}{K}\sum_{(h,w)\in\mathcal{T}_K} a_{h,w}$, where $\mathcal{T}_K$ denotes the set of $K = 3$ patches with the highest anomaly scores. This top-K strategy is particularly suited for manufacturing data where anomalies are locally concentrated, as it prevents the dilution of anomalous signals by surrounding normal regions.

5 Experiments

5.1 Experimental Setup

Datasets and Protocol We evaluate DeCoFlow on MVTec-AD [2] and VisA [42] benchmarks. All experiments follow the target 1×1 CIL scenario: classes are learned sequentially under zero-replay constraints and task-agnostic inference. To ensure consistent comparison, we apply an identical alphabetical learning order for all methods across both benchmarks.

Baselines and Metrics We compare against joint-training (PatchCore [27], CADIC [34]), fine-tuning (PatchCore, CFA [13], SimpleNet [20], RD4AD [5]), and diverse CL baselines (EWC [11], LwF [16], Replay, IUF [32], ReplayCAD [10], CDAD [15], DNE [14], UCAD [19], CADIC [34]). Primary evaluation metrics are I-AUC and P-AP, with FM (↓) and routing accuracy as auxiliary measures.

Implementation Details For MVTec-AD, features are extracted from a WideResNet-50-2 backbone. The NF is configured with 6 DCL and 2 ACL blocks at LoRA rank 16, then trained via AdamP (lr=3×10^{-4}) for 60 epochs with a batch size of 16. For VisA, we utilize ViT-B/16 features from blocks [1,2,3,5] with 10 DCL and 6 ACL blocks (lr=2×10^{-4}) for 100 epochs. Both benchmarks train the base on Task 0, freeze it, and subsequently update only the adapters. Results denote the mean and standard deviation over three seeds at 224×224 resolution; full-stream training takes about 3.0 GPU-hours on MVTec-AD and 4.1–4.4 GPU-hours on VisA.

5.2 Main Results

MVTec-AD DeCoFlow achieves 98.40% I-AUC, surpassing the joint reference baselines in image-level detection, with competitive 58.20% P-AP. It maintains

Table 2: Performance comparison on MVTec-AD and VisA in the 1×1 CIL setting. Entries denote average performance (Avg) and forgetting measure (FM) after the final task. **Bold**/underlined values indicate best/second-best results. *Joint* denotes reference baselines trained on all tasks simultaneously. † signifies parameter-level zero forgetting under correct routing; ‡ results are from [34]. All values are in %.

Method	MVTec-AD (15-class)				VisA (12-class)			
	I-AUC		P-AP		I-AUC		P-AP	
	Avg (↑)	FM (↓)	Avg (↑)	FM (↓)	Avg (↑)	FM (↓)	Avg (↑)	FM (↓)
Joint_PatchCore	*97.80*	–	*59.40*	–	*91.60*	–	*44.00*	–
Joint_CADIC	*97.80*	–	*59.10*	–	*91.00*	–	*43.90*	–
FT_PatchCore‡	60.20	38.30	19.00	37.10	58.90	36.10	9.00	31.10
FT_CFA‡ [13]	62.30	36.10	17.70	8.30	59.30	32.70	8.70	18.40
FT_SimpleNet‡ [20]	70.80	21.10	6.00	6.90	61.60	28.30	1.40	1.60
FT_RD4AD‡ [5]	59.60	39.30	14.30	42.50	52.50	42.30	6.90	20.10
IUF [32]	76.20	6.70	17.10	5.90	68.10	8.50	3.40	0.30
ReplayCAD [10]	94.80	4.50	53.70	5.50	90.30	5.50	41.50	5.00
CDAD [15]	74.30	20.10	28.00	26.10	62.80	22.20	8.30	21.70
DNE [14]	87.00	11.60	–	–	61.00	17.90	–	–
UCAD [19]	93.00	1.00	45.60	1.30	87.40	3.90	30.00	1.50
CADIC [34]	97.20	1.10	**58.40**	1.50	89.10	4.30	**43.80**	1.40
DeCoFlow (Ours)	**98.40**±0.00	**0.00**†	58.20±0.10	**0.00**†	**93.00**±0.10	**0.00**†	37.00±0.30	**0.00**†

parameter-level zero forgetting (FM=0.00%) by structurally bypassing inter-task parameter interference (Corollary 1): unlike replay- or regularization-based methods, DeCoFlow secures stability through structural parameter isolation rather than constrained updates. On MVTec-AD, Mahalanobis routing also reaches 100% accuracy. Performance is stable across class orders, with alphabetical, reverse, and random streams yielding I-AUC 98.47/98.43/98.40 and P-AP 58.57/59.20/58.23 at FM=0.00%.

VisA On VisA, DeCoFlow attains 93.00% I-AUC with zero forgetting, outperforming ReplayCAD by +2.70%p and CADIC by +3.90%p in detection. Pixel-level localization is weaker, with P-AP at 37.00% against CADIC's 43.80%, and the gap concentrates on fine-defect classes such as *chewinggum* (ΔP-AP −51.3) and *pcb3* (−28.8), where thin, low-contrast boundaries are oversmoothed (Fig. 2). This behavior traces to feature granularity rather than the density model: the ViT-B/16 backbone raises I-AUC over WideResNet-50-2 (93.00% vs. 88.60%) but yields a coarse 14×14 grid, and raising token resolution lifts P-AP from 35.85% to 39.52%, while stronger backbones reach I-AUC/P-AP 95.77/44.01 (EVA02-B14) and P-AP 38.05 (ViT-B/8). Detection thus remains robust, with localization the primary VisA limitation.

Qualitative Evidence To complement quantitative metrics, Fig. 4 provides qualitative anomaly maps. Across representative classes, DeCoFlow assigns low scores to normal regions and concentrates high responses on defect areas, consistent with the image-level gains in Tab. 2.

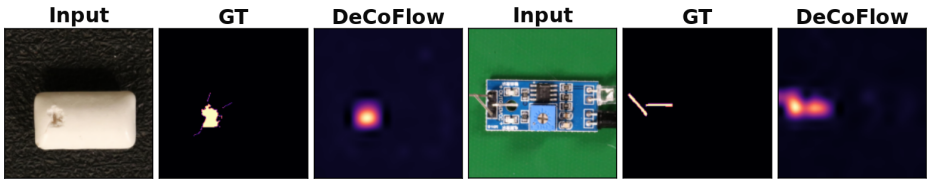

Fig. 2: VisA fine-defect localization failures (Input, GT, DeCoFlow). Defect regions are detected, but thin or low-contrast boundaries are blurred—consistent with the fine-defect P-AP gap rather than a density-modeling failure.

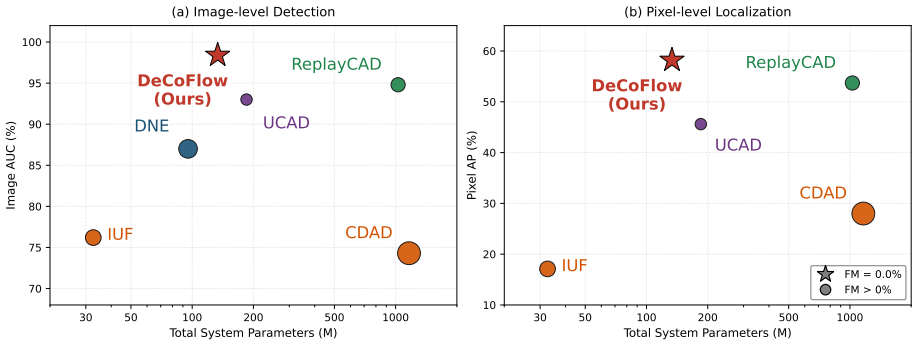

Fig. 3: Performance, parameter cost, and forgetting comparison on MVTec-AD 15-class.

Structural Cost Analysis Figure 3 positions DeCoFlow against baselines in the accuracy–parameter–forgetting trade-off. Table 3 confirms that updating the shared base (up to 25.0× cost) or employing high-rank adapters (4.7–7.5× overhead) yields no significant gains. DeCoFlow adopts rank=16, aligned with the SVD effective rank of 17.23 (Tab. 6a), requiring only 2.27M parameters per task (2.5%). End-to-end inference takes 90.2 ms/image (11.1 FPS) with 989 MB peak memory on the 15-task MVTec-AD stream, faster than CADIC's 10k-coreset setting (6.6 FPS) while storing parameters rather than coresets.

5.3 Ablation Study

Components Analysis Coupling decomposition alone (without TSA and TAL) achieves 94.18%, establishing DCL as the primary architectural component (Tab. 4). The performance gap between this baseline and the full model reflects frozen-base rigidity, which each auxiliary module mitigates through a specialized function. Specifically, removing ACL incurs the largest degradation (−10.17%p), indicating that DCL requires explicit normalization to resolve residual statistical mismatches (Fig. 5). Similarly, the absence of TAL causes underfitting in low-density tail regions (−3.17%p), while removing TSA compromises feature alignment for subsequent tasks (−0.68%p). Finally, excluding LoRA results in a negligible ac-

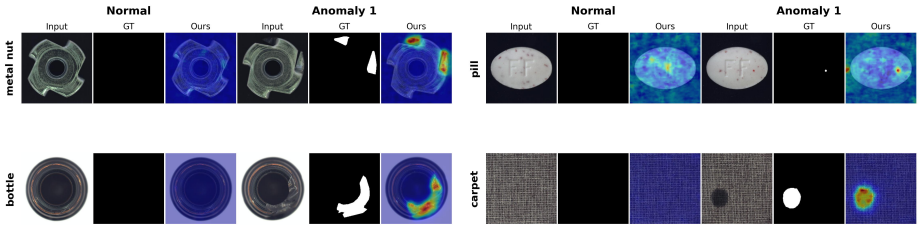

Fig. 4: Qualitative anomaly maps for four representative classes (one normal and two anomalous samples per class). DeCoFlow assigns consistently low scores to normal regions and localized high responses to defect regions.

Table 3: Design space comparison (MVTec-AD): varying adapter structure and configuration on the same NF backbone. Rel. denotes parameter ratio relative to DeCoFlow.

Design	Adapter	I-AUC (%)	P-AP (%)	Params (M)	Rel.
Full ACL (8b, h=0.5)	Full/task	98.74	58.27	7.10	3.1×
Full ACL (8b, h=2.0)	Full/task	98.83	57.81	56.68	25.0×
DCL + Linear	Full-rank	98.48	57.78	10.64	4.7×
DCL + LoRA (r=512)	LoRA	98.46	58.56	17.12	7.5×
DCL-only 6b (w/o ACL)	LoRA (r=16)	88.30	45.76	1.49	0.66×
DCL-only 8b (w/o ACL)	LoRA (r=16)	89.46	47.38	2.12	0.93×
DeCoFlow (6DCL+2ACL)	**LoRA** (r=16)	**98.49**	**58.57**	**2.27**	**1.0×**

Table 4: Component ablation results (MVTec-AD). ‡w/o LoRA entails 100% parameter increase per task.

Configuration	I-AUC (%)	P-AP (%)	Δ I-AUC
Full (DeCoFlow)	98.47	58.57	–
w/o TSA + TAL	94.18	48.56	−4.29
w/o ACL	88.30	45.76	−10.17
w/o TAL	95.30	49.67	−3.17
w/o TSA	97.79	55.00	−0.68
w/o LoRA‡	98.43	58.90	−0.04

curacy impact (−0.04%p) but doubles the parameter count per task, confirming its role as a critical efficiency mechanism.

Regime-Specific Interaction Analysis Factorial analysis in Tab. 5 reveals that our compensators are specifically tailored for the frozen NF base regime rather than being universally beneficial modules. TAL provides substantial gains (+13.04%p) when the NF base weights are frozen but becomes redundant or even deleterious (−1.22%p) when the base is trainable, validating its specialized role as a capacity reallocation mechanism. While ACL improves performance in both settings, its contribution is 1.66× greater under a frozen NF base. TAL is also locally stable: TailTopK/TailW sweeps vary I-AUC only within 97.83–97.92% and P-AP within 56.03–56.18%. Finally, TSA contributes primarily through synergistic interaction with other modules rather than through standalone gains.

5.4 Mechanism Analysis

Table 6 analyzes the principles behind DeCoFlow's precise task adaptation with few parameters from two perspectives.

Low-Rank Sufficiency SVD analysis of ΔW after 15-task training (Tab. 6a) reveals a mean effective rank of 17.23. Our configuration of rank 16 directly aligns with this inherent low-dimensional structure, capturing the essential sig-

Table 5: Factorial analysis: I-AUC change (%p) per component under frozen vs. trainable base regimes. Δ is measured from each regime's uncompensated baseline (Frozen: 84.12%, Trainable: 68.13%). 2×4 design to characterize regime-dependent behavior.

Component	Frozen (Δ%p)	Trainable (Δ%p)	Regime Dep.
TAL	+13.04	−1.22	Frozen-specific
ACL	+11.52	+6.93	1.66× amplified
TSA	−1.19	−16.34	Suppressed

Table 6: Mechanism analysis. Part A: Low-rank sufficiency (24 ΔW matrices, 15-task average). Part B: TAL gradient redistribution (4 tasks × 30 batches average). Part C: Task-agnostic base validation (5 different Task 0 initializations).

Part A: Low-Rank Sufficiency

Metric	Value	Perf.
Eff. Rank (SVD)	17.23±1.10	—
Energy@16	92.82%	98.49 / 58.57
Energy@64	99.71%	98.47 / 58.57
RAR ($\|\Delta W\|/\|W\|$)	0.092±0.027	—

Part B: TAL Gradient Redistribution

Loss Config	Grad (Tail)	Grad (Non-Tail)	Ratio
Mean-only	0.00296	0.00212	1.38×
Tail-Aware	0.1104	0.000755	144.2×
Amplification	37.1×	0.36×	104.5×

Part C: Task-Agnostic Base

Metric	NF	MLP
Cos-Sim	**0.78**±0.06	0.27±0.11
I-AUC Std	0.21%p	—
Routing	100%	—

nal required for SOTA performance with minimal overhead. This sufficiency is further supported by the high inter-task gradient similarity (cos-sim 0.78, Tab. 6c), which suggests that task adaptation occurs within a compact subspace. Additionally, the relative adaptation ratio (RAR = $\|\Delta W\|_F / \|W_{\text{base}}\|_F$) averages only 0.092, confirming that the frozen base handles 90.8% of the overall transformation while the low-rank adapters manage task-specific adjustments. A rank sweep from r=16 to 128 changes I-AUC/P-AP by only 0.02/0.09%p, indicating that performance is governed more by the DCL/ACL boundary and TAL objective than by raw adapter rank.

Tail Gradient Amplification Under the frozen-base regime, we leverage the limited capacity of low-rank adapters through strategic gradient allocation rather than relying on parameter-intensive full-rank updates. Standard NLL optimization produces a near-uniform gradient distribution, with a tail-to-non-tail ratio of only 1.38× (Tab. 6b). Conversely, TAL significantly amplifies this ratio to **144.2**×, representing a 104.5-fold increase. This shift effectively concentrates the available parameter budget on critical low-density boundaries where normal and anomalous distributions overlap. This redistribution enables a +3.17%p I-AUC gain (Tab. 4) without overhead, showing targeted optimization can outperform capacity scaling in continual learning.

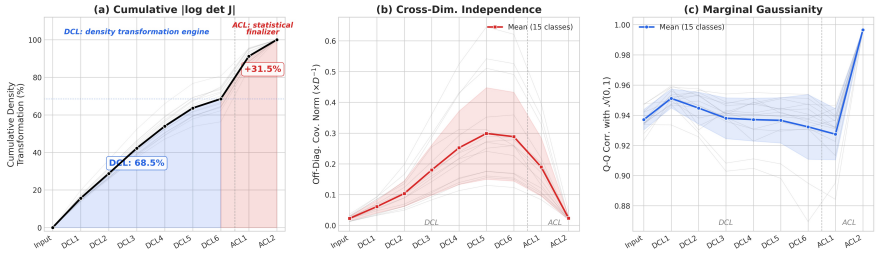

Fig. 5: Block-wise transformation analysis. (a) Cumulative $|\log|\det \mathbf{J}_f(\mathbf{z})||$: DCL 6 blocks contribute 68.5%, ACL 2 blocks contribute 31.5%. (b) Off-Diagonal Covariance: DCL introduces cross-dimensional coupling, which ACL restores to independence. (c) Q-Q Correlation: stagnates through DCL, then sharply aligns after ACL. Gray lines denote individual classes (15), bold line denotes the mean.

Block-Wise Complementarity Fig. 5 illustrates how DCL and ACL operate in tandem to achieve precise density modeling. While DCL serves as the primary engine by executing 68.5% of the total transformation (Fig. 5a), it inevitably induces cross-dimensional coupling, as evidenced by a rise in off-diagonal covariance (Fig. 5b). ACL subsequently acts as a statistical finalizer, restoring independence and ensuring sharp marginal Gaussian alignment (Fig. 5c). This complementarity is not substitutable: replacing ACL with additional DCL blocks (8DCL+0ACL) yields a 9.01%p performance deficit despite maintaining identical depth. These results confirm that DCL handles the heavy lifting of transformation while ACL provides the necessary statistical regularization for accurate anomaly detection.

Task-Agnostic Base Validation The frozen-base strategy assumes that the initial Task 0 base generalizes across all subsequent tasks (Tab. 6c). To validate this, we independently trained Task 0 on five different classes and measured the pairwise cosine similarity of their resulting gradients. The average similarity reached **0.78**, indicating highly aligned optimization trajectories across different classes. As a control, a standard MLP yielded a similarity of only 0.27 ($p < 0.003$, Welch's t-test), confirming that this alignment is an intrinsic property of the affine coupling architecture when paired with a shared Gaussian target. Furthermore, varying the initial Task 0 class resulted in a final I-AUC standard deviation of only **0.21%p**. These results demonstrate that DeCoFlow's effectiveness is invariant to the choice of the initial task, providing strong empirical support for the frozen-base approach.

6 Conclusion

We formalize subnet independence in affine coupling layers as a mechanism for parameter isolation in continual learning. DeCoFlow decomposes Normalizing Flows into a frozen base and task-specific low-rank adapters, structurally eliminating parameter-interference forgetting under correct routing, while TSA, ACL,

and TAL compensate for frozen-base rigidity to deliver $FM = 0.00\%$ on the evaluated task-agnostic streams with state-of-the-art detection on MVTec-AD and VisA.

Beyond accuracy, our analysis shows that coupling-internal decomposition is uniquely effective in preventing manifold collapse, and that DCL and ACL play complementary roles—dominant density transformation versus statistical normalization—elucidating the mechanism behind the adapter modules.

Limitations remain. The VisA localization gap motivates higher-resolution/multi-scale density modeling and stronger representations, and a diagnostic 4×4 mixed-CIL test on VisA shows lower stability when multiple normal distributions are mixed from the first task (I-AUC 77.97%, P-AP 21.43%). Since end-to-end forgetting is ultimately mediated by routing, the guarantee weakens as prototypes overlap: under far more similar classes or much longer streams, routing rather than parameter isolation becomes the limiting factor. Strengthening routing in such regimes, broader protocols (mixed $N \times M$, task-/domain-incremental, large-scale, and cross-domain CAD), prototype routing cost, and extensions to other generative models remain future work. More broadly, leveraging architecture-specific structural invariance for parameter isolation offers a principled path toward forgetting-free continual learning beyond normalizing flows.

Acknowledgements. This work was supported by Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (RS-2024-00460011, Climate and Environmental Data Platform for Enhancing Climate Technology Capabilities in the Anthropocene (CEDP)) and (RS2021-II211343, Artificial Intelligence Graduate School Program (Seoul National University)). This work was also supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (RS-2024-00407803, RS-2025-23523657). This work was supported by the Korea Planning & Evaluation Institute of Industrial Technology (KEIT), funded by the Ministry of Trade, Industry and Energy (MOTIE), Korea (No. RS-2026-25509027, Development and Demonstration of an Ontology- and On-Device AI-Based Autonomous Operation Agent System for Semiconductor Wafer Manufacturing Processes). This work was also supported by the BK21 FOUR Program (Education and Research Center for Industrial Innovation Analytics) funded by the Ministry of Education, Korea (No. 4120240214912).

References

1. Aljundi, R., Babiloni, F., Elhoseiny, M., Rohrbach, M., Tuytelaars, T.: Memory aware synapses: Learning what (not) to forget. In: European Conference on Computer Vision (ECCV). pp. 139–154 (2018)
2. Bergmann, P., Fauser, M., Sattlegger, D., Steger, C.: Mvtec ad–a comprehensive real-world dataset for unsupervised anomaly detection. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 9592–9600 (2019)

3. Cheng, Q., Wan, Y., Wu, L., Hou, C., Zhang, L.: Continuous subspace optimization for continual learning. In: Advances in Neural Information Processing Systems (2025)
4. De Lange, M., Aljundi, R., Masana, M., Parisot, S., Jia, X., Leonardis, A., Slabaugh, G., Tuytelaars, T.: A continual learning survey: Defying forgetting in classification tasks. IEEE Transactions on Pattern Analysis and Machine Intelligence **44**(7), 3366–3385 (2022). https://doi.org/10.1109/TPAMI.2021.3057446
5. Deng, H., Li, X.: Anomaly detection via reverse distillation from one-class embedding. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9737–9746 (2022)
6. Dinh, L., Sohl-Dickstein, J., Bengio, S.: Density estimation using real-nvp. In: International Conference on Learning Representations (2017)
7. French, R.M.: Catastrophic forgetting in connectionist networks. Trends in Cognitive Sciences **3**(4), 128–135 (1999)
8. Gudovskiy, D., Ishizaka, S., Kozuka, K.: Cflow-ad: Real-time unsupervised anomaly detection with localization via conditional normalizing flows. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 98–107 (2022)
9. Hu, E.J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., Chen, W.: Lora: Low-rank adaptation of large language models. In: International Conference on Learning Representations (2022)
10. Hu, L., Gan, Z., Deng, L., Liang, J., Liang, L., Huang, S., Chen, T.: Replaycad: Generative diffusion replay for continual anomaly detection. arXiv preprint arXiv:2505.06603 (2025)
11. Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A.A., Milan, K., Quan, J., Ramalho, T., Grabska-Barwinska, A., et al.: Overcoming catastrophic forgetting in neural networks. Proceedings of the national academy of sciences **114**(13), 3521–3526 (2017)
12. Lee, A., Zhang, Y., Gomes, H.M., Bifet, A., Pfahringer, B.: Look at me, no replay! SurpriseNet: Anomaly detection inspired class incremental learning. In: International Conference on Information and Knowledge Management (2023)
13. Lee, S., Lee, S., Song, B.C.: CFA: Coupled-hypersphere-based feature adaptation for target-oriented anomaly localization. IEEE Access **10**, 78446–78454 (2022). https://doi.org/10.1109/ACCESS.2022.3193699
14. Li, C.L., Lin, T.H., Tsai, Y.Y., Lin, Y.C., Yang, M.H., Kira, Z.: Towards continual adaptation in industrial anomaly detection. In: Proceedings of the 30th ACM International Conference on Multimedia (2022)
15. Li, X., Tan, X., Chen, Z., Zhang, Z., Zhang, R., Guo, R., Jiang, G., Chen, Y., Qu, Y., Ma, L., Xie, Y.: One-for-more: Continual diffusion model for anomaly detection. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2025)
16. Li, Z., Hoiem, D.: Learning without forgetting. In: European conference on computer vision. pp. 614–629. Springer (2017)
17. Liang, Y.S., Chen, J.R., Li, W.J.: Gated integration of low-rank adaptation for continual learning of large language models. In: Advances in Neural Information Processing Systems (2025)
18. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 2980–2988 (2017)

19. Liu, J., Wang, K., Jiang, Q., Feng, Z., Zhang, W., You, Z.: Unsupervised continual anomaly detection with contrastively-learned prompt. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 38 (2024)
20. Liu, Z., Zhou, Y., Xu, Y., Wang, Z.: Simplenet: A simple network for image anomaly detection and localization. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 20402–20411 (2023)
21. Lopez-Paz, D., Ranzato, M.: Gradient episodic memory for continual learning. In: Advances in Neural Information Processing Systems (2017)
22. Mallya, A., Lazebnik, S.: PackNet: Adding multiple tasks to a single network by iterative pruning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 7765–7773 (2018)
23. McCloskey, M., Cohen, N.J.: Catastrophic interference in connectionist networks: The sequential learning problem. In: Psychology of Learning and Motivation, vol. 24, pp. 109–165. Academic Press (1989)
24. Momeni, S., Noroozi, V., et al.: AnaCP: Toward upper-bound continual learning via analytic contrastive projection. In: Advances in Neural Information Processing Systems (2025)
25. Pang, J., Li, C.: Context-aware feature reconstruction for class-incremental anomaly detection and localization. Neural Networks **181**, 106788 (2025). https://doi.org/10.1016/j.neunet.2024.106788
26. Qiu, Z., Xu, Y., He, C., Meng, F., Xu, L., Wu, Q., Li, H.: MINGLE: Mixtures of null-space gated low-rank experts for test-time continual model merging. In: Advances in Neural Information Processing Systems (2025)
27. Roth, K., Pemula, L., Zepeda, J., Schölkopf, B., Brox, T., Gehler, P.: Towards total recall in industrial anomaly detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 14318–14328 (2022)
28. Rusu, A.A., Rabinowitz, N.C., Desjardins, G., Soyer, H., Kirkpatrick, J., Kavukcuoglu, K., Pascanu, R., Hadsell, R.: Progressive neural networks. arXiv preprint arXiv:1606.04671 (2016)
29. Serrà, J., Surís, D., Miron, M., Karatzoglou, A.: Overcoming catastrophic forgetting with hard attention to the task. In: International Conference on Machine Learning. pp. 4548–4557. PMLR (2018)
30. Shin, H., Lee, J.K., Kim, J., Kim, J.: Continual learning with deep generative replay. In: Advances in Neural Information Processing Systems. pp. 2990–2999 (2017)
31. Shrivastava, A., Gupta, A., Girshick, R.: Training region-based object detectors with online hard example mining. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 761–769 (2016)
32. Tang, J., Lu, H., Xu, X., Wu, R., Hu, S., Zhang, T., Cheng, T.W., Ge, M., Chen, Y.C., Tsung, F.: An incremental unified framework for small defect inspection. In: European Conference on Computer Vision (ECCV). pp. 306–323. Springer (2024)
33. Wang, X., Zhang, R., Liang, Y.S., Li, W.J.: C-LoRA: Continual low-rank adaptation for pre-trained models. arXiv preprint arXiv:2502.17920 (2025)
34. Yang, G., Deng, Z., Man, J.: CADIC: Continual anomaly detection based on incremental coreset. arXiv preprint arXiv:2511.08634 (2025)
35. Yao, X., Li, R., Qian, Z., Wang, L., Zhang, C.: Hierarchical gaussian mixture normalizing flow modeling for unified anomaly detection. In: European Conference on Computer Vision (ECCV). pp. 93–110. Springer (2024)
36. You, Z., Cui, L., Shen, Y., Yang, K., Lu, X., Zheng, Y., Le, X.: A unified model for multi-class anomaly detection. In: Advances in Neural Information Processing Systems (2022)

37. Yu, J., Zheng, Y., Wang, X., Li, W., Wu, Y., Zhao, R., Wu, L.: Fastflow: Unsupervised anomaly detection and localization via 2d normalizing flows. arXiv preprint arXiv:2111.07677 (2021)
38. Zhao, Y., Zhu, K., Shi, H., Guo, Y., Bai, C.: OmniAL: A unified cnn framework for unsupervised anomaly localization. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3924–3933 (2023)
39. Zhou, Y., Xu, X., Song, J., Shen, F., Shen, H.T.: MSFlow: Multiscale flow-based framework for unsupervised anomaly detection. IEEE Transactions on Neural Networks and Learning Systems (2024). https://doi.org/10.1109/TNNLS.2023.3346541
40. Zhou, Y., Xu, X., Song, J., Shen, F., Shen, H.T.: VQ-Flow: Taming normalizing flows for multi-class anomaly detection via hierarchical vector quantization. arXiv preprint arXiv:2409.00942 (2024)
41. Zhou, Y., Li, X., Ding, J.: Multimodal task representation memory bank vs. catastrophic forgetting in anomaly detection. arXiv preprint arXiv:2502.06194 (2025)
42. Zou, Y., Jeong, J., Pemula, L., Zhang, D., Dabeer, O.: Spot-the-difference self-supervised pre-training for anomaly detection and segmentation. In: European Conference on Computer Vision. pp. 392–408. Springer (2022)