# D-MECAD: Dynamic Multi-Expert Continual Anomaly Detection with Adaptive Architecture Discovery

Malihe Dahmardeh[1] and Francesco Setti[1,2]

[1] Department of Engineering for Innovation Medicine, University of Verona, Italy
[2] Qualyco S.r.l., strada le Grazie 15, Verona, Italy
Corresponding Author: malihe.dahmardeh@univr.it

**Abstract.** In this paper, we propose Dynamic Multi-Expert Continual Anomaly Detection (D-MECAD), a self-organizing framework that autonomously discovers the optimal number of expert networks for sequential anomaly detection tasks. The framework employs an Adaptive Assignment mechanism that dynamically transitions from exploratory expert creation to strategic consolidation, regulating expert creation through feature similarity and capacity constraints. A novel Progressive Monitoring protocol continuously evaluates assignment quality using held-out validation data, identifying victim classes (experiencing performance degradation) and toxic classes (destabilizing expert performance), triggering automatic reassignment to preserve knowledge. Evaluation on the MVTec AD dataset shows that the system autonomously converges to 7 experts, achieving an average Area Under the Receiver Operating Characteristic curve (AUROC) of 0.94 with both cosine similarity and Euclidean distance metrics. Validation across similarity metrics demonstrates robustness of the dynamic discovery mechanism, while generalizability testing on the VisA dataset confirms autonomous expert allocation without manual architecture tuning. Results establish an effective plasticity-stability balance in dynamic industrial environments.

**Keywords:** Multi-Expert · Continual Learning · Anomaly Detection.

## 1 Introduction

Anomaly Detection (AD) has become increasingly relevant in industrial applications, where detecting defects and irregularities in manufacturing processes is critical for quality control and operational efficiency. Traditional AD methods, particularly those based on supervised learning, require extensive labeled data, which can be expensive and time-consuming to collect. As a result, unsupervised AD approaches have gained traction, allowing systems to identify anomalies without relying on explicitly labeled abnormal samples.

One of the primary limitations of existing AD models is that they are trained on specific object categories and cannot generalize well when deployed in dynamic environments where new objects and variations frequently emerge. In industrial production lines, for example, manufacturing conditions, product types, and defect patterns change rapidly, necessitating models that can adapt to new object classes efficiently without requiring retraining from scratch. This adaptability is crucial to ensuring continued performance without incurring the high cost of re-annotation and re-optimization of the dataset and the model.

Continual learning (CL) provides a promising approach to address these challenges. By enabling models to learn sequentially from new tasks while preserving knowledge from previously learned tasks, CL mitigates catastrophic forgetting and allows for long-term adaptation. However, applying CL to anomaly detection poses unique challenges, as conventional CL techniques are typically designed for classification rather than unsupervised tasks. Many existing CL strategies require extensive retraining or memory buffers to store past data, making them impractical for real-time industrial applications.

Recent multi-expert architectures for continual anomaly detection have demonstrated the value of specialized knowledge distribution across independent expert networks. However, these approaches rely on predefined expert configurations, require manual specification of the number of experts, and lack mechanisms to adapt when initial class assignments prove suboptimal. A critical gap exists in developing systems that can autonomously discover the appropriate expert architecture while continuously monitoring and validating assignment decisions.

In this work, we extend our previous work [7] to D-MECAD, a self-organizing framework that autonomously determines both the number and the specialization of expert networks for continual anomaly detection. Our system advances beyond fixed-architecture approaches through two key innovations. First, an Adaptive Assignment mechanism dynamically regulates expert creation by transitioning from exploratory building to strategic consolidation, using dual similarity thresholds and capacity constraints to discover the optimal expert configuration without manual specification. Second, a Progressive Monitoring protocol continuously validates assignment quality using held-out validation data during sequential learning, automatically detecting victim classes and toxic classes and triggering targeted reassignment without full model retraining. Evaluation on standard benchmarks demonstrates that our framework autonomously converges to compact expert architectures while achieving substantial performance improvements over fixed-configuration baselines, with consistent behavior across multiple similarity metrics and datasets.

The main contributions of our paper are as follows:

- Autonomous Expert Discovery: A self-organizing framework that dynamically determines the optimal number of specialized expert networks through adaptive similarity-based assignment, eliminating manual architecture while mitigating catastrophic forgetting through specialized knowledge distribution.
- Progressive Monitoring with Validation-Guided Correction: A protocol that continuously validates assignment quality using held-out data during sequential learning, detecting performance degradation and triggering surgical reassignment of victim and toxic classes without full model retraining.

– We demonstrate framework generalizability of consistent autonomous expert discovery and performance through validation across multiple similarity metrics (cosine, Euclidean) and datasets, showing robust autonomous expert discovery and consistent performance independent of metric choice or domain characteristics.

## 2 Related Work

### 2.1 Unsupervised Anomaly Detection

Anomaly detection (AD) methods aim to identify patterns deviating from normal data distributions and can be broadly categorized into reconstruction-based and feature-embedding-based techniques. Reconstruction-based methods such as (VAEs) [15] and (GANs) [32] identify anomalies based on high reconstruction errors, while feature-embedding methods including memory bank models [9,28] and one-class SVMs [10] eparate normal and anomalous samples in latent space. Recent transformer-based approaches MoEAD [24] and ADMoE [39] introduce mixture-of-experts for anomaly scoring but do not address continual learning. Our work builds on PatchCore [28] toward continual anomaly detection through specialized expert networks.

### 2.2 Continual Learning

Continual learning (CL) focuses on sequential learning of multiple tasks without catastrophic forgetting. Regularization-based methods such as(EWC) [1,16,25,38] constrain weight updates to preserve prior task knowledge but struggle with highly divergent tasks. Replay methods [26,27]maintain a memory buffer of past samples, with coreset selection strategies [4,2,34], improving sample informativeness. Expert-based methods [3,19] train specialized models per task; [12,17,29,36].

Recent theoretical analysis MoE-Adapters [35]for vision language models and dual-momentum strategies [14] for multimodal learning. TAME [40] employs a task detection mechanism to activate the most relevant expert based on loss deviation. However, these approaches typically focus on classification rather than unsupervised anomaly detection. Our approach combines replay-based principles with the multi-expert architecture for the specific challenges of continual anomaly detection, where the goal is to identify anomalies in sequentially introduced object classes rather than classify them.

### 2.3 Continual Learning for Anomaly Detection (CLAD)

Recent works have begun to address continual learning in anomaly detection contexts specifically, demonstrating how structured memory retention enhances knowledge transfer and reduces forgetting [18]. A benchmark for continual anomaly detection with pixel-level evaluation has also been introduced [6]. Proposed approaches span multiple paradigms. Traditional strategies include fitting of normal data distributions [20], adaptive learning [8] and incremental learning [30], continual prompting and contrastive learning within a Vision Transformer [22], and reverse distillation [33]. Recent diffusion-based approaches [13,21] coreset-based methods [31] have further advanced the field. These works have demonstrated that well-designed replay and memory management strategies significantly outperform conventional approaches in dynamic environments [11]. Contemporary methods such as EVOLVE [36], ReplayCAD [13], and One-for-More [21] advance the field, but they rely on fixed model architectures or task-specific retraining.

Nevertheless, they often face trade-offs between adaptability, accuracy, and computational efficiency. Existing methods typically use fixed model architectures or lack explicit mechanisms for managing expertise across diverse object classes. Our approach addresses these specific gaps by leveraging a dynamic multi-expert architecture with specialized memory management techniques, offering a more effective framework for handling the dynamic nature of industrial anomaly detection tasks compared to the methods discussed above.

## 3 Multi-expert continual anomaly detection

We consider a continual anomaly detection setting where a system observes a sequential stream of object classes $\mathcal{C} = \{c_1, c_2, \dots, c_T\}$, introduced one at a time. At step $t$, only anomaly-free training images for class $c_t$ are available; raw data from previous steps is inaccessible, reflecting industrial constraints where storing full past datasets is impractical. At inference, the system receives a test image and routes it through the expert assigned to its class during training, producing an image-level anomaly score. Performance is measured by AUROC averaged across all seen classes. Catastrophic forgetting is quantified as Backward Transfer (BWT) [23], the average AUROC change on previously seen classes after new classes are introduced, where values near zero indicate stable knowledge retention.

Unlike conventional multi-expert approaches with predefined configurations, D-MECAD dynamically determines both the number of experts and their class assignments through an Adaptive Assignment mechanism and progressive validation correction protocol. The overall pipeline is presented in Figure 1.

### 3.1 Multi-Expert Architecture

Our model maintains a dynamically growing set of experts $E = \{E_1, E_2, ..., E_K\}$, where $K$ is discovered autonomously during sequential learning rather than predetermined. Each expert $E_i$ specializes in one or more object classes through similarity-based assignment. Following PatchCore [28], experts operate as independent memory banks storing patch-level embeddings of normal samples, sharing a fixed feature extractor to ensure consistent representations across all experts. The number of experts emerges from the interplay between Adaptive Assignment mechanism based on feature similarity and monitoring performance by validation assignment protocol, balancing specialization against computational efficiency.

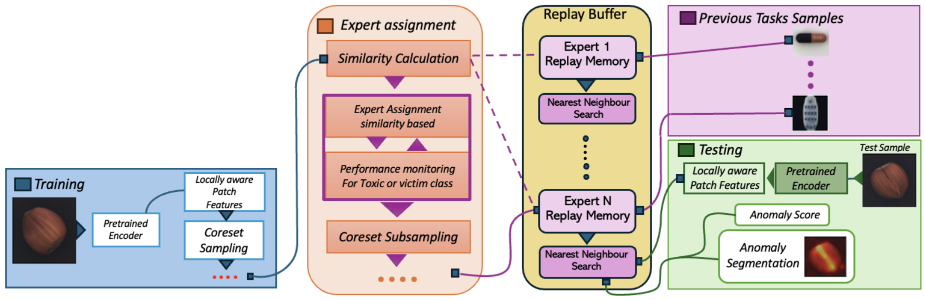



Fig. 1: Overview of the D-MECAD pipeline. The system processes object classes sequentially, extracting patch-level embeddings from normal samples, Coreset of these embeddings are assigned to specialized experts based on cosine similarity. Each expert maintains a memory buffer with randomly selected samples and is updated only when assigned a new class. During inference, test samples are processed through the corresponding expert to generate anomaly scores at the image level.

### 3.2 Patch embedding memory-bank

Our framework relies on an efficient patch-level memory representation for anomaly detection. This approach is inspired by PatchCore's mechanism [28], but introduces novel mechanisms specifically designed for the continual learning context. Given an input image $I \in \mathbb{R}^{H \times W \times 3}$, we extract discriminative patch-level features from intermediate layers of a pre-trained WideResNet50 [37] backbone. During training, we build a memory bank of normal patch embeddings .For efficient resource utilization, we apply Coreset selection [34] to reduce the full set of patch embeddings into a compact yet representative memory bank.For anomaly detection at inference time, test image patches are compared against this memory bank using a nearest-neighbor approach. Our multi-expert architecture extends this memory-bank approach by maintaining separate, specialized banks for different experts, each focusing on specific object classes based on feature similarity.

### 3.3 Adaptive Expert Assignment with Continuous Monitoring

Our assignment strategy operates as a continuous loop that processes each arriving class through assignment, validation for potential reassignment, training and evaluation. This iterative process enables the system to discover optimal expert configurations while continuously monitor suboptimal assignments.

*Class Assignment Process:* When a new class $c$ arrives, we determine its expert assignment through similarity-based routing. For the first class, we create the first expert $E_1$ and assign it directly. For all subsequent classes, we compute the similarity between the class centroid $\mu_c$ and each existing expert's centroid $\mu_{E_i}$:

$$\mathrm{sim}(c, E_i) = \frac{\langle \mu_c, \mu_{E_i} \rangle}{\|\mu_c\| \cdot \|\mu_{E_i}\|} \tag{1}$$

for cosine similarity, or $\mathrm{sim}(c, E_i) = -\|\mu_c - \mu_{E_i}\|_2$ for Euclidean distance. The assignment decision for class c follows:

$$\mathcal{A}(c) = \begin{cases} \arg\max_i \mathrm{sim}(c, E_i) & \text{if } \max_i \mathrm{sim}(c, E_i) \geq \theta \text{ and } |\mathcal{C}_{E_i}| < M \\ E_{\text{new}} & \text{otherwise} \end{cases} \tag{2}$$

where $K$ is the current number of experts, $j$ is the maturity threshold for number of experts, $s = \max_i \mathrm{sim}(c, E_i)$ is the maximum similarity score, $M$ is the maximum capacity per expert, and $|\mathcal{C}_{E_i}|$ denotes the number of classes in expert $E_i$ and $\theta$ is similarity threshold.

The assignment strategy adapts based on the system's maturity. In the early exploration phase (where $K < j$), we apply only a primary similarity threshold $\theta_1$ to encourage initial expert specialization and discovery. Once the system matures ($K \geq j$), we introduce a secondary relaxed threshold $\theta_2$ (where $\theta_2 < \theta_1$) to enable strategic allocation. During this phase, candidate classes exhibiting similarity scores $s$ such that $\theta_2 \leq s < \theta_1$ are assigned to existing experts rather than triggering creation of new ones. This dual-threshold mechanism balances fine-grained specialization and computational efficiency without manual architecture specification.

*Progressive Performance Monitoring:* Immediately after assigning class $c$ to expert $E_i$, we establish performance baselines monitoring for degradation. To enable rigorous assignment quality assessment during sequential learning without contaminating final evaluation, we partition each class's test data into two disjoint sets, a validation set reserved for all progressive monitoring decisions, and a held-out test set used exclusively for final system evaluation. This separation ensures that test data is never accessed during the continual learning process.

After training expert $E_i$ with the newly assigned class $c$, we evaluate its validation performance and record:

$$\mathrm{AUROC}_{\text{baseline}}(c, E_i) = \mathrm{Eval}_{\text{val}}(E_i, c) \tag{3}$$

This baseline represents the achievable performance for class $c$ in expert $E_i$ immediately after assignment. For existing classes $j \in \mathcal{C}_{E_i} \setminus \{c\}$ already in expert $E_i$, we retrieve their stored original baselines that were established when they were first assigned to this expert.

*Performance Degradation Detection:* We then perform comprehensive validation evaluation of all existing classes to detect potential performance degradation caused by the newly added class. For each existing class $j \in \mathcal{C}_{E_i} \setminus \{c\}$, we measure current validation performance:

$$\mathrm{AUROC}_{\text{current}}(j, E_i) = \mathrm{Eval}_{\text{val}}(E_i, j), \tag{4}$$

and we compute performance degradation by comparing against the stored original baseline:

$$\Delta_j = \text{AUROC}_{\text{baseline}}(j, E_i) - \text{AUROC}_{\text{current}}(j, E_i). \tag{5}$$

*Victim and Toxic Class Detection:* We compare current performance against baselines to identify problematic assignments. A class $j \in \mathcal{C}_{E_i} \setminus \{c\}$ is flagged as a victim, i.e. it suffers from negative interference caused by the new class, if $\Delta_j > \tau_{\text{victim}}$, where $\tau_{\text{victim}}$ is a fixed threshold. To identify toxic classes that cause widespread degradation, we examine whether the newly added class $c$ damages all existing classes. Class $c$ is flagged as toxic if $\Delta_j > \tau_{\text{victim}} \quad \forall j \in \mathcal{C}_{E_i} \setminus \{c\}$. That is, if every existing class experiences performance drops exceeding $\tau_{\text{victim}}$, then class $c$ is toxic and its presence destabilizes the entire expert. When multiple victims are detected —i.e., more than one class suffers degradation, but not all classes— we treat this as a multi-victim scenario and handle the newly added class $c$ as toxic, isolating it to prevent further interference.

*Surgical Reassignment:* Upon detecting victims and toxic classes, we immediately perform surgical correction using validation-based assessments. Victim reassignment consists of removing the victim class $j$ from expert $E_i$ and assigning it to the expert with the highest similarity that has not reached maximum capacity, with validation performance on the target expert guiding the choice. If no suitable expert exists, a new dedicated expert is created for class $j$. Toxic isolation instead removes the toxic class $c$ from expert $E_i$ and places it in a new dedicated expert, after which $E_i$ is retrained without class $c$ using replay samples from the remaining classes. Finally, memory reconstruction is performed for both source and destination experts by rebuilding their memory banks using the updated class assignments from replay buffers, ensuring clean separation after reassignment.

*Iterative Refinement* After surgical correction, we re-evaluate all affected experts on validation data to verify that reassignment has resolved performance issues. This evaluation-correction loop continues until no victims or toxic classes remain, ensuring stable expert configurations before proceeding to the next class.

### 3.4 Expert Training with Replay

We then train the assigned expert $E_i$ by combining the new class embeddings with replay samples from previously assigned classes:

$$\mathcal{M}_{E_i} = \text{Coreset}\left(\mathcal{S}_c \cup \{r \in \mathcal{R}_{E_i,j} : j \in \mathcal{C}_{E_i} \setminus \{c\}\}\right) \tag{6}$$

where $\mathcal{S}_c$ contains all patch embeddings from class $c$, $\mathcal{R}_{E_i,j}$ is the replay buffer for class $j$,

### 3.5 Anomaly Detection Inference

During testing, for an image from class $c$, we extract patch embeddings and process them through the corresponding expert $E_i$ that was assigned to class $c$ during training. For image-level anomaly detection, we compute the maximum patch-level anomaly score:

$$s_{\text{image}} = \max_{p \in P} s(p, \mathcal{M}_{E_i}) \tag{7}$$

where $P$ is the set of patch embeddings $p$ from the test image and $s(p, \mathcal{M}_{E_i})$ measures the distance between patch $p$ and the nearest neighbor in the expert's memory $\mathcal{M}_{E_i}$. This architecture ensures that experts remain specialized and isolated during inference, with no feature mixing across experts.

Our framework achieves effective continual learning for anomaly detection by combining expert specialization, strategic class assignment, and efficient memory management. The experimental results demonstrate that this approach successfully mitigates catastrophic forgetting while maintaining high anomaly detection performance across sequentially introduced object classes.

## 4 Experiments

We evaluate D-MECAD on MVTec AD [5], and VisA [41], analyzing expert discovery, per-class performance, forgetting mitigation, and replay buffer effectiveness across similarity metrics.

### 4.1 Experimental Setup

We implemented our framework using PyTorch with Amazon's PatchCore as the backbone architecture. For feature extraction, we used a WideResNet50 pre-trained on ImageNet, extracting features from layers 2 and 3. Input image size of $224 \times 224$ pixels and Patch size of $32 \times 32$ pixels. All experiments were conducted on a single NVIDIA GPU with 16GB of memory. The training process was performed sequentially, with classes introduced in the same order across all experimental configurations to ensure fair comparison. In the assignment strategy the primary threshold $\theta_1$=0.9 enforces high-confidence grouping in cosine space $[-1, 1]$, ensuring only genuinely similar classes share an expert. The relaxed threshold $\theta_2$=0.8 enables consolidation without forcing unnecessary expert creation. The maturity threshold $j$=3 represents the minimum number of experts before meaningful cross-expert similarity patterns emerge. The victim detection threshold $\tau$=0.1 corresponds to a 10-point AUROC drop, consistent with forgetting thresholds in prior CL-AD work [11].

*Datasets.* MVTec-AD [5] is a standard benchmark for industrial anomaly detection, consisting of 15 object categories with 5,354 high-resolution images. Each category contains normal (defect-free) samples for training and both normal and anomalous samples for testing. We validate generalization on the VisA dataset [41] with 12 classes covering complex assembly and food products. Classes are introduced in a fixed order across all experiments; all methods use the identical ordering to ensure fair comparison. MVTec AD and VisA represent contrasting continual learning challenges: MVTec presents high inter-class diversity across heterogeneous object categories, while VisA provides semantically coherent classes within PCB inspection and food quality domains.

*Evaluation Protocol and Metrics.* We adopt a continual learning protocol in which object classes are introduced sequentially. Performance is evaluated using image-level AUROC (Area Under the Receiver Operating Characteristic curve) for anomaly detection, along with Backward Transfer (BWT) [23], which measures the average change in AUROC on previously seen classes after each new class is introduced and serves as a proxy for forgetting (values close to zero indicate stable retention). We also report memory usage, capturing both the distribution of classes across experts and the storage requirements associated with each.

*Baselines.* We compare D-MECAD against three reference configurations that bracket the performance space. **Single-expert MECAD** [7] trains one shared memory bank on all classes sequentially without expert routing, representing naive continual learning with maximum catastrophic forgetting (0.7494 AUROC, BWT $= -0.374$ on MVTec AD). **Fixed 5-expert MECAD** [7] uses our predecessor architecture with a manually specified expert count and static assignment, without dynamic discovery or progressive monitoring (0.8259 AUROC, BWT $= -0.140$). **Per-class PatchCore** trains one independent PatchCore model per class with no continual learning constraint, representing the performance upper bound at the cost of $T$ independent models.

### 4.2 Results and Analysis

We present comprehensive results demonstrating the autonomous expert discovery capability of D-MECAD across different similarity metrics. Compared to the single-expert lower bound (0.7494 AUROC, BWT $= -0.374$) and the fixed 5-expert MECAD baseline [7] (0.8259 AUROC, BWT $= -0.140$), D-MECAD achieves 0.9404 AUROC with 7 autonomously discovered experts while maintaining near-zero forgetting (BWT $= -0.003$), substantially closing the gap toward the per-class upper bound. Table 1 summarizes autonomous discovery across datasets and similarity metrics. The framework demonstrates adaptive scaling, discovering 7 experts for MVTec AD's diverse 15 classes versus 3-5 experts for VisA's semantically coherent 12 classes. Strong performance maintained across domains (MVTec: 0.94, VisA: 0.89), with VisA's lower scores attributable to challenging classes (capsules, macaroni2) paralleling MVTec's screw difficulty.

Table 1: Autonomous expert discovery across datasets and similarity metrics. Expert count adapts to dataset characteristics without manual specification.

| Dataset | Metric | Classes | Experts | AUROC |
|---|---|---|---|---|
| MVTec AD | Euclidean | 15 | 7 | 0.9422 |
| | Cosine | 15 | 7 | 0.9404 |
| VisA | Euclidean | 12 | 5 | 0.8861 |
| | Cosine | 12 | 3 | 0.8877 |

**Per-Expert and Per-Class Performance Analysis** MVTec AD Results, Figure 2 presents the cosine similarity configuration with 7 autonomously discovered experts. Expert-level performance ranges from 0.8548 (Expert 4, containing the challenging screw class) to 1.0000 (Expert 3, isolated leather), demonstrating effective specialization. The framework successfully grouped semantically coherent classes while isolating toxic ones—metal_nut (Expert 5) and wood (Expert 7) were surgically isolated after destabilizing multi-class experts during initial assignment. Notably, Expert 4 accommodates 4 classes including the challenging screw (0.5324 AUROC), demonstrating cosine similarity's scale-invariant properties enable coexistence despite performance disparities. Per-class analysis reveals 11 of 15 classes achieved AUROC $\geq 0.95$, with only screw (0.5324), macaroni1 (0.8085), and two moderately performing classes falling below this threshold.

In comparison, the Euclidean distance configuration also converged to 7 experts but required stricter isolation: screw (0.5499) and toothbrush (1.0000) were both surgically isolated due to toxic detection, whereas cosine's scale-invariance allowed toothbrush to coexist successfully in Expert 4. Expert-level performance ranged from 0.5499 (isolated screw) to 0.9905 (Expert 1, 2 classes), with similar overall patterns but greater fragmentation.

VisA Dataset Results, Figure 4 demonstrates generalization to the VisA dataset, where cosine similarity autonomously discovered 3 experts with perfectly balanced allocation—each expert contains exactly 4 classes. Expert performance ranges from 0.8438 (Expert 0) to 0.9296 (Expert 2), showing tighter distribution compared to MVTec due to greater semantic coherence. Notably, no toxic classes were detected, validating the effectiveness of similarity-based grouping for semantically related product categories. The challenging capsules class (0.6757) remained the lowest-performing across both datasets, while PCB4 achieved the highest performance (0.9872). The four PCB variants distributed across all three experts rather than clustering together, demonstrating the system's sensitivity to feature-space heterogeneity beyond visual similarity.

In contrast, Euclidean distance for VisA converged to 5 experts and triggered toxic isolation for PCB3 (Expert 4, 0.8899 AUROC), reflecting greater sensitivity to inter-class interference. This cross-metric comparison validates that cosine similarity consistently provides superior architectural efficiency (fewer experts) and stability, achieving equivalent or better performance through scale-invariant similarity computation that enables more effective class consolidation.

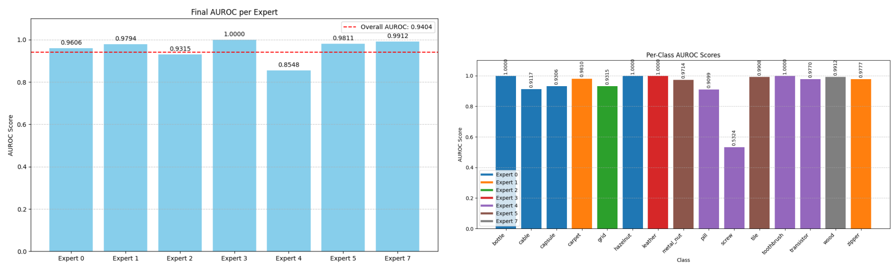



Fig. 2: Cosine similarity results on MVTec AD dataset: (a) Per-expert AUROC scores , (b) Per-class AUROC distribution.

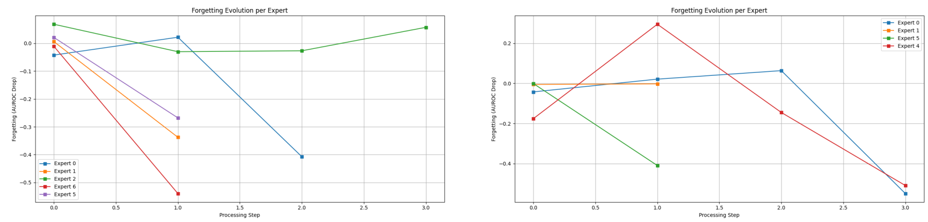
Table 2: Toxic class detection and surgical isolation across datasets and metrics.

| **Dataset** | **Metric** | **Class** | **Action** | **Final AUROC** |
|---|---|---|---|---|
| MVTec | Euclidean | screw | Toxic isolation (Expert 4) | 0.5499 |
| | | toothbrush | Toxic isolation (Expert 6) | 1.0000 |
| | Cosine | metal_nut | Toxic isolation (Expert 5) | 0.9714 |
| | | wood | Toxic isolation (Expert 7) | 0.9912 |
| VisA | Euclidean | pcb3 | Toxic isolation (Expert 4) | 0.8899 |
| | Cosine | – | No toxic detection | – |

The four PCB variants (pcb1-4) distributed across experts rather than clustering together, with Euclidean detecting pcb3 as toxic to other PCBs (Expert 2). This validates the system's sensitivity to feature-space heterogeneity beyond visual similarity. PCB4 achieved highest performance (0.9901-0.9905) across both metrics. Capsules (0.6541) emerged as requiring isolation in both metrics and exhibiting the lowest per-class AUROC.

**Progressive Performance Monitoring Effectiveness** The continuous monitoring loop actively intervened during sequential learning to prevent catastrophic forgetting. Table 2 summarizes detected toxic classes and surgical isolation actions. The protocol successfully prevented performance degradation by detecting toxic classes through baseline validation comparison ($\Delta$AUROC $> 0.10$).

MVTec triggered 4 toxic isolations (2 per metric), while VisA triggered only 1 (Euclidean), reflecting MVTec's greater inter-class diversity requiring protective isolation. Toxic PCB3 in VisA parallels screw/toothbrush in MVTec—mechanically complex classes destabilizing similarity-based groupings. In Euclidean configuration, both screw and toothbrush were permanently isolated upon detection. In cosine configuration, metal_nut and wood were both isolated to dedicated experts, with both maintaining strong final performance (metal_nut: 0.9714, wood: 0.9912). This demonstrates the system's ability to protect against interference through surgical isolation. Notably, cosine's scale invariance enabled




(a) Forgetting evolution per expert with cosine similarity

(b) Forgetting evolution per expert with Euclidean distance

Fig. 3: Evaluation of forgetting per expert (a) with Cosine Similarity and (b) with Euclidean Distance, for each expert in the 7-expert configuration as classes are introduced sequentially

the challenging screw class to coexist with other classes in Expert 4 (achieving 0.5324 AUROC), whereas Euclidean required complete isolation for the same class (0.5499 AUROC). The surgical correction protocol successfully prevented catastrophic forgetting across all detected toxic cases.

**Memory Budget Utilization** Our framework enforced strict per-class (5000 samples) and per-expert (25000 samples) memory budgets. Expert memory utilization in the final configurations for MVTec AD and Visa datasets shows that Experts ranged from 20% (single-class) to 80% utilization (4-class) demonstrating efficient capacity exploitation.

**Forgetting Analysis** Figure 3 illustrates forgetting evolution as new classes are introduced. For Euclidean Distance 3b, multi-class experts demonstrate stable forgetting: Expert 0 (4 classes) maintains 0.05-0.10 after initial fluctuations, while Expert 1 (2 classes) shows minimal forgetting ($<0.05$) from semantic coherence. Single-class experts (2, 3, 6, 7) maintain zero forgetting by design. Expert 4's forgetting spike corresponds to toxic screw isolation, where surgical removal prevented catastrophic degradation. For Cosine Similarity 3a, Expert 0 exhibits consistently lower forgetting ($<0.05$) across 4 classes compared to Euclidean, while Expert 4 (5 classes post-wood consolidation) shows controlled forgetting around 0.10, validating the consolidation threshold.

**Limitations.** Our framework assumes a brief class setup phase with validation samples available, consistent with industrial inspection workflows where new product classes are introduced under controlled conditions. Following standard practice in memory-bank anomaly detection [28,9], we employ a frozen WideResNet50 backbone, trading feature adaptability for stable cross-expert similarity measurements. Class centroids are computed from initial training samples and not updated over time; centroid drift in long-horizon deployments is a direction for future work.

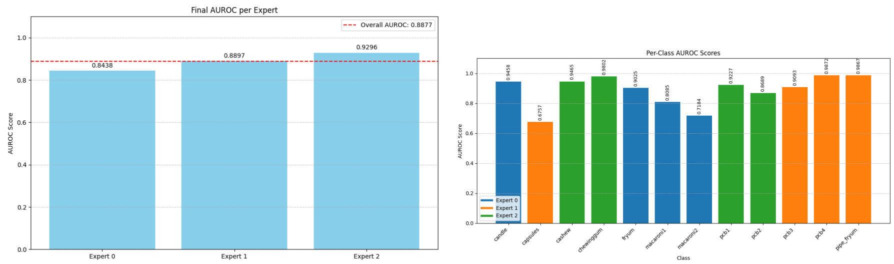



Fig. 4: VisA with cosine similarity results: (a) Balanced 4-class-per-expert allocation, (b) PCB variants distributed across experts based on feature heterogeneity.

## 5 Conclusion

We presented D-MECAD, a self-organizing framework for continual anomaly detection that autonomously discovers optimal expert architectures through adaptive assignment and progressive performance monitoring. Evaluation on MVTec AD and VisA datasets demonstrates autonomous convergence to compact configurations (7 experts for MVTec's 15 classes, 3-5 for VisA's 12 classes) achieving 0.94 and 0.89 AUROC respectively, with forgetting $\leq 0.10$ across 13 of 15 classes. Cross-metric validation reveals cosine similarity provides superior architectural efficiency and stability compared to Euclidean distance due to scale-invariant properties. The progressive monitoring protocol successfully prevented catastrophic forgetting through validation-based surgical corrections, establishing a foundation for adaptive anomaly detection in dynamic industrial environments without manual architecture tuning.

## 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 (2018)
2. Aljundi, R., Caccia, L., Belilovsky, E., Caccia, M., Lin, M., Charlin, L., Tuytelaars, T.: Online continual learning with maximally interfered retrieval. In: International Conference on Neural Information Processing Systems (2019)
3. Aljundi, R., Chakravarty, P., Tuytelaars, T.: Expert gate: Lifelong learning with a network of experts. In: IEEE Conference on Computer Vision and Pattern Recognition (2017)
4. Aljundi, R., Lin, M., Goujaud, B., Bengio, Y.: Gradient based sample selection for online continual learning. Advances in Neural Information Processing Systems **32** (2019)
5. Bergmann, P., Fauser, M., Sattlegger, D., Steger, C.: MVTec AD–a comprehensive real-world dataset for unsupervised anomaly detection. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (2019)
6. Bugarin, N., Bugaric, J., Barusco, M., Pezze, D.D., Susto, G.A.: Unveiling the anomalies in an ever-changing world: A benchmark for pixel-level anomaly detection

in continual learning. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (2024)
7. Dahmardeh, M., Setti, F.: Mecad: A multi-expert architecture for continual anomaly detection. arXiv preprint arXiv:2512.15323 (2025)
8. Dalle Pezze, D., Anello, E., Masiero, C., Susto, G.A.: Continual learning approaches for anomaly detection. arXiv preprint arXiv:2212.11192 (2022)
9. Defard, T., Setkov, A., Loesch, A., Audigier, R.: Padim: a patch distribution modeling framework for anomaly detection and localization. In: International Conference on Pattern Recognition (2021)
10. Deng, H., Li, X.: Anomaly detection via reverse distillation from one-class embedding. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (2022)
11. Faber, K., Corizzo, R., Sniezynski, B., Japkowicz, N.: Lifelong continual learning for anomaly detection: New challenges, perspectives, and insights. IEEE Access **12**, 41364–41380 (2024)
12. Hihn, H., Braun, D.A.: Mixture-of-variational-experts for continual learning. arXiv preprint arXiv:2110.12667 (2021)
13. Hu, L., Gan, Z., Deng, L., Liang, J., Liang, L.: Replaycad: Generative diffusion replay for continual anomaly detection. In: Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence (IJCAI-25) (2025)
14. Huai, T., Zhou, J., Wu, X., Chen, Q., Bai, Q., Zhou, Z., He, L.: Cl-moe: Enhancing multimodal large language model with dual momentum mixture-of-experts for continual visual question answering. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2025)
15. Kingma, D.P.: Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114 (2013)
16. 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)
17. Le, M., Nguyen, A., Nguyen, H., Nguyen, T., Pham, T., Van Ngo, L., Ho, N.: Mixture of experts meets prompt-based continual learning. arXiv preprint arXiv:2405.14124 (2024)
18. Lee, J.C., Kim, T., Park, E., Woo, S.S., Ko, J.H.: Continuous memory representation for anomaly detection. In: European Conference on Computer Vision (2025)
19. Li, H., Lin, S., Duan, L., Liang, Y., Shroff, N.B.: Theory on mixture-of-experts in continual learning. arXiv preprint arXiv:2406.16437 (2024)
20. Li, W., Zhan, J., Wang, J., Xia, B., Gao, B.B., Liu, J., Wang, C., Zheng, F.: Towards continual adaptation in industrial anomaly detection. In: ACM International Conference on Multimedia (2022)
21. 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)
22. Liu, J., Wu, K., Nie, Q., Chen, Y., Gao, B.B., Liu, Y., Wang, J., Wang, C., Zheng, F.: Unsupervised continual anomaly detection with contrastively-learned prompt. In: AAAI Conference on Artificial Intelligence (2024)
23. Lopez-Paz, D., Ranzato, M.: Gradient episodic memory for continual learning. In: Advances in Neural Information Processing Systems (2017)
24. Meng, S., Meng, W., Zhou, Q., Li, S., Hou, W., He, S.: Moead: A parameter-efficient model for multi-class anomaly detection. In: European Conference on Computer Vision (2025)

25. Nguyen, C.V., Li, Y., Bui, T.D., Turner, R.E.: Variational continual learning. arXiv preprint arXiv:1710.10628 (2017)
26. Rebuffi, S.A., Kolesnikov, A., Sperl, G., Lampert, C.H.: icarl: Incremental classifier and representation learning. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (2017)
27. Rolnick, R.D., Ahuja, A., Schwarz, J., Lillicrap, T.P., Wayne, G.: Experience replay for continual learning. arXiv:1811.11682 (2018)
28. Roth, K., Pemula, L., Zepeda, J., Schölkopf, B., Brox, T., Gehler, P.: Towards total recall in industrial anomaly detection. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (2022)
29. Rypeść, G., Cygert, S., Khan, V., Trzciński, T., Zieliński, B., Twardowski, B.: Divide and not forget: Ensemble of selectively trained experts in continual learning. arXiv preprint arXiv:2401.10191 (2024)
30. 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 (2025)
31. Tran, T.A., et al.: Cadic: Continual anomaly detection based on incremental coreset. arXiv preprint arXiv:2511.08634 (2025)
32. Yan, X., Zhang, H., Xu, X., Hu, X., Heng, P.A.: Learning semantic context from normal samples for unsupervised anomaly detection. In: AAAI Conference on Artificial Intelligence (2021)
33. Yang, A., Xu, X., Wu, Y., Liu, H.: Reverse distillation for continuous anomaly detection. IEEE Transactions on Instrumentation and Measurement (2024)
34. Yoon, J., Madaan, D., Yang, E., Hwang, S.J.: Online coreset selection for rehearsal-based continual learning. In: International Conference on Learning Representations (ICLR) (2021)
35. Yu, J., Zhang, Y., Lu, J., Xu, Y., Yan, K., Shen, Y., Xing, E.P.: Boosting continual learning of vision-language models via mixture-of-experts adapters. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2024)
36. Yu, X., Rosing, T., Guo, Y.: Evolve: Enhancing unsupervised continual learning with multiple experts. In: IEEE/CVF Winter Conference on Applications of Computer Vision (2024)
37. Zagoruyko, S., Komodakis, N.: Wide residual networks. arXiv preprint arXiv:1605.07146 (2016)
38. Zenke, F., Poole, B., Ganguli, S.: Continual learning through synaptic intelligence. In: International Conference on Machine Learning (2017)
39. Zhao, Y., Zheng, G., Mukherjee, S., McCann, R., Awadallah, A.: Admoe: Anomaly detection with mixture-of-experts from noisy labels. In: AAAI Conference on Artificial Intelligence (2023)
40. Zhu, H., Majzoubi, M., Jain, A., Choromanska, A.: Tame: Task agnostic continual learning using multiple experts. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (2024)
41. 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 (2022)