Memory-Augmented Autoencoder for Industrial Anomaly Detection

Hai Son[1], Duc Ngo[2*]
Faculty of Information Technology, Posts and Telecommunications Institute of Technology, Hanoi, Vietnam[1, 2]
Laboratory for Advanced Networking Applications, Posts and Telecommunications Institute of Technology, Hanoi, Vietnam[2]

***Abstract*—Unsupervised anomaly detection in industrial manufacturing requires identifying defective products utilizing only defect-free training images. Reconstruction-based methods using autoencoders are widely applied but suffer from limited discriminative power when defects occupy small image regions. This work introduces a Memory-Augmented Autoencoder (MAA) that combines a U-Net reconstruction component with a patch-level memory bank for nearest-neighbor-based anomaly scoring. During inference, MAA computes a hybrid anomaly score from both the reconstruction error and the memory distance to produce image-level predictions and spatial anomaly heatmaps. Evaluated on the bottle category of the MVTec AD benchmark, MAA attains an image-level AUC-ROC of 0.985 and an Average Precision of 0.996, surpassing all autoencoder-based baselines including SSIM AE (0.930) and VAE (0.910) and performing on par with the pretrained PaDiM method using a ResNet-18 backbone (0.982), while relying on only 168 defect-free training images and no ImageNet pretraining. A pixel-level AUC-ROC of 0.895 indicates a reasonable spatial localization capability. Ablation studies demonstrate that the memory-based component is the primary driver of detection performance, outperforming the reconstruction component by 16.4 percentage points in AUC-ROC, which is attributable to the robustness of patch-level memory matching for localized structural defects. As the present evaluation is confined to a single object category, the reported results are best interpreted as a proof of concept rather than as evidence of broad generalization.**

Keywords*—*Anomaly detection; autoencoder; memory bank; nearest-neighbor search; MVTec AD; unsupervised learning; industrial inspection; U-Net

I. Introduction

Automated visual inspection is a critical component of quality control in large-scale industrial manufacturing. The goal is to identify defective products as early as possible in the production pipeline to minimize waste and ensure product reliability. A fundamental challenge in this setting is the cold-start problem [1]: defect-free (nominal) images are abundant and easily acquired, but defective examples are rare, diverse, and difficult to enumerate exhaustively. This precludes supervised classification approaches and motivates the formulation of anomaly detection as an unsupervised, one-class classification problem.

Existing approaches to unsupervised industrial anomaly detection span several paradigms. Reconstruction-based methods [2, 3, 4] train autoencoders to reconstruct normal images, operating under the assumption that an autoencoder trained exclusively on normal data will restore normal test images well but fail on anomalous ones. However, this assumption is often violated in practice: powerful decoder architectures, particularly those with skip connections, can reproduce anomalous regions nearly as well as normal ones, suppressing the reconstruction error signal. Generative adversarial approaches [5] and normalizing flows [6] have been explored as alternatives, but often suffer from training instability and high sensitivity to hyperparameter choices. More recently, methods relying on ImageNet pretrained features [1, 7] have achieved state-of-the-art (SOTA) performance by matching test patch features against a memory bank of nominal features using nearest-neighbor search. These approaches consistently outperform reconstruction-based methods, yet they often require large pretrained backbone networks and access to vast external training corpora, limiting their applicability in resource-constrained or domain-specific settings.

This work proposes the Memory-Augmented Autoencoder (MAA), a lightweight approach that combines the complementary strengths of reconstruction-based and memory-based detection within a single, self-contained framework trained exclusively on target-domain images. MAA consists of three components: 1) a U-Net encoder-decoder that learns to reconstruct normal images and produces a compact latent feature representation, 2) a patch-level memory bank constructed from encoder features of the training set, supporting efficient nearest-neighbor-based anomaly scoring at inference time, and 3) a hybrid scoring module that combines reconstruction error and memory distance into a final image-level anomaly score. The memory bank component fundamentally addresses the reconstruction shortcut problem: even when the decoder can reconstruct anomalous regions, the encoder features of those regions remain out-of-distribution with respect to the stored normal patches, thereby preserving the anomaly signal.

MAA is evaluated on the bottle category of the MVTec Anomaly Detection (MVTec AD) benchmark [8], one of the most widely used datasets for industrial anomaly detection. The main contributions of this work are:

- A memory-augmented autoencoder architecture that integrates patch-level memory matching with reconstruction-based detection without requiring any pretrained backbone or external data.
- An ablation study demonstrating that the memory component outperforms the reconstruction component by 16.4 percentage points in AUC-ROC, providing quantitative evidence for the reconstruction shortcut problem in skip-connection architectures.

*Corresponding author.

- An achievement of competitive performance with pretrained SOTA methods on the MVTec AD bottle category (AUC-ROC = 0.9849), achieved with only 168 training images, a 3.1M parameter network, and CPU-only inference.
- A detailed analysis of per-defect-type detection behavior across three defect categories (*broken_large, broken_small,* and *contamination*), demonstrating the advantage of patch-level memory scoring for small, localized defects.

The remainder of this study is organized as follows. Section II reviews related work. Section III describes the MAA architecture and training procedure. Section IV presents experimental results. Section V concludes with a discussion of limitations and future directions.

II. Related Work

A. Reconstruction-Based Anomaly Detection

Autoencoder-based anomaly detection methods train an encoder–decoder network on normal images and use pixel-level reconstruction error as an anomaly score at test time [2]. Bergmann et al. [8] provide a systematic evaluation of autoencoder variants on the MVTec AD benchmark, including a vanilla L2 autoencoder (Vanilla AE, AUC-ROC = 0.860 on bottle), a variational autoencoder (VAE, AUC-ROC = 0.910), and an SSIM-based autoencoder (SSIM AE, AUC-ROC = 0.930) that incorporates a structural similarity loss [9] to capture perceptual differences beyond per-pixel intensity. While SSIM AE improves upon pixel-level baselines by penalizing structural deviations, these methods share a common limitation: reconstruction error is computed globally over the entire image, diluting local anomaly signals when defects occupy only a small fraction of the image area.

A fundamental problem affecting U-Net architectures specifically is the *reconstruction shortcut*: skip connections allow the decoder to directly access high-resolution encoder feature maps, enabling near-perfect reconstruction of anomalous regions by copying texture information and bypassing the bottleneck. This effect is particularly pronounced for texture categories where the repeated surface patterns can be transmitted through skip connections without meaningful compression. In the MAA framework, this limitation is addressed by decoupling detection from reconstruction: the memory bank operates on the bottleneck encoder features, which do not benefit from skip connections and therefore retain anomaly sensitivity.

B. Memory-Based Anomaly Detection

Memory-based approaches for anomaly detection store representations of normal data at training time and score test samples by their distance to the nearest stored representation. SPADE [10] constructs a multi-scale memory bank from ImageNet pretrained features and uses *k*-nearest-neighbor (*k*-NN) search for both image-level detection and pixel-level segmentation. PaDiM [7] estimates a multivariate Gaussian distribution at each patch location in the feature map and uses Mahalanobis distance for anomaly scoring, achieving strong localization by accounting for feature correlations. PatchCore [1] extends the memory bank paradigm by introducing locally-aware, neighborhood-aggregated mid-level patch features combined with greedy coreset subsampling for efficient retrieval, achieving near-perfect performance on MVTec AD (mean AUC-ROC 0.991) while retaining fast inference times.

A key limitation shared by SPADE, PaDiM, and PatchCore is their dependence on ImageNet pretrained backbone networks, which provide rich semantic features but introduce a potential distribution shift between natural image statistics and industrial inspection targets. PatchCore [1] explicitly acknowledges this, motivating the use of mid-level rather than deep features to reduce ImageNet classification bias. In contrast, MAA trains its encoder directly on the target domain, producing features specifically adapted to the normal appearance of the inspected product, with a trade-off in feature richness compared to large pretrained models.

Closest in spirit to MAA is the Memory-Augmented Autoencoder proposed by Gong et al. [11] for video anomaly detection. Their approach augments a convolutional autoencoder with a memory module operating in the latent space, trained end-to-end with a memory sparsity constraint. MAA differs in three key respects: 1) it utilizes a U-Net architecture with skip connections for reconstruction quality, 2) the memory bank is constructed post-training from encoder features without end-to-end joint optimization, and 3) it employs standard *k*-NN distance rather than memory addressing, which is more interpretable and easier to tune.

C. Hybrid Approaches

Several works have explored combinations of reconstruction and feature-matching signals. DRAEM [12] trains a discriminative network with synthetic anomalies generated by pasting random textures onto normal images, achieving strong performance on texture categories by learning an explicit normal/anomaly boundary. Unlike MAA, DRAEM relies on externally synthesized anomalies (from the Describable Textures Dataset [13]) during training, which introduces a dependency on the anomaly simulation quality. MAA instead targets a purely unsupervised setting with no access to anomaly examples, real or synthetic.

D. Recent Advances in Industrial Anomaly Detection

Research on industrial anomaly detection has advanced rapidly in recent years; Liu et al. [14] provide a comprehensive review organized by network architecture, level of supervision, and benchmark protocol. Several of these directions bear directly on the design choices made in MAA.

Density-estimation methods model the distribution of nominal features explicitly. FastFlow [15] applies two-dimensional normalizing flows as a plug-in probability estimator on top of an arbitrary pretrained backbone, jointly capturing local and global feature relations. Knowledge-distillation methods instead exploit the representation discrepancy between a teacher and a student network. Reverse distillation [16] inverts the usual data flow so that a student decoder restores the multi-scale representations of a teacher encoder from a one-class embedding, which limits the tendency of the student to generalize to anomalous inputs. EfficientAD [17] carries this paradigm towards real-time operation using a

shallow patch description network together with a loss that discourages the student from imitating the teacher beyond the nominal distribution, reporting millisecond-level latencies. These methods are complementary to MAA in that they retain a pretrained teacher, whereas MAA derives all of its features from the target domain.

Feature-adaptation methods are closest to the motivation of the present work. CFA [18] argues that the features of a pretrained CNN remain biased towards their source domain and introduces a learnable patch descriptor together with a scalable memory bank so that nominal features become more compact for the target dataset. SimpleNet [19] similarly inserts a shallow feature adapter to transfer pretrained features towards the target domain and trains a discriminator against synthetic anomalies generated in feature space. Both corroborate the concern that motivates MAA, namely that ImageNet features are not automatically well matched to industrial inspection data, yet both still depend on a pretrained backbone. Transformer backbones have also been examined in this setting: VT-ADL [20] combines a vision transformer encoder with a Gaussian mixture density estimator for joint detection and localization.

Relative to these methods, MAA occupies a deliberately conservative point in the design space. It forgoes pretrained features, synthetic anomalies, and flow-based or distillation-based machinery, and instead examines how far a plain U-Net encoder combined with a patch-level memory bank can be taken when only a few hundred nominal images from the target domain are available. The comparison in Section IV-G should therefore be read as situating a deliberately minimal baseline among substantially heavier methods rather than as a like-for-like contest.

III. Methodology

The MAA framework consists of three components: a U-Net autoencoder for reconstruction (Section III-A), a patch-level memory bank for nearest-neighbor scoring (Section III-B), and a hybrid anomaly scoring module (Section III-C). Fig. 1 illustrates the U-Net autoencoder architecture, and Fig. 2 details the two-stream inference pipeline.

A. U-Net Autoencoder

The backbone of MAA is a U-Net encoder-decoder with depth 4 and base filter count 32. The encoder consists of four successive convolutional blocks, each comprising two 3×3 convolutions with ReLU activations and batch normalization, followed by 2×2 max-pooling. The filter counts double at each depth level: 32, 64, 128, 256. The final encoder block produces a feature map of shape 16×16×128, which serves as the latent representation. The decoder mirrors the encoder with four transposed-convolution up-sampling stages, each concatenated with the corresponding encoder feature map via skip connections before being processed by two 3×3 convolution layers.

The network is trained to minimize a hybrid reconstruction loss combining mean squared error (MSE) and the structural similarity index (SSIM) [9]:

$$L_{hybrid} = 0.7 \times L_{MSE} + 0.3 \times (1 - SSIM(x, \hat{x})) \quad (1)$$

where, x denotes the input image and $\hat{x}$ denotes the reconstruction. The MSE term enforces per-pixel fidelity while the SSIM term penalizes structural deviations in local image patches, providing a perceptually more meaningful reconstruction objective. The weighting coefficient 0.7 was selected to balance the two terms empirically.

Training uses the Adam optimizer [21] with an initial learning rate of 1×10^{-4} and batch size 8 for a maximum of 50 epochs. Two callbacks regulate training: *EarlyStopping* with patience 10 that restores the best observed weights, and *ReduceLROnPlateau* with reduction factor 0.5 and patience 5 (minimum $lr = 1\times10^{-7}$). All images are resized to 256×256 pixels and normalized to [0, 1].

B. Patch-Level Memory Bank

After training the autoencoder, a memory bank is constructed from the encoder features of all training images. For each training image, the method extracts the 16×16×128 latent feature map and partitions it into overlapping 3×3 patches with stride 1, yielding $(16-3+1)^2 = 196$ patches per image. Each patch is flattened to a vector of dimensionality 3×3×128 = 1,152. For a training set of 168 images, this produces a memory bank M of 168×196 = 32,928 patch vectors:

$$M = \{\varphi(x_i, h, w) \mid x_i \in X_N, h, w \in \{1, \dots, 14\}\} \quad (2)$$

where, $\varphi(x_i, h, w)$ denotes the 3×3 patch feature centered at position (h, w) in the encoder output for training image x_i. The memory bank is indexed using a Ball Tree structure for efficient nearest-neighbor retrieval. At inference, the anomaly score for a test patch p_{test} is the Euclidean distance to its nearest neighbor in M:

$$s_{mem}(p_{test}) = \min_{m \in M} \|p_{test} - m\|_2 \quad (3)$$

The image-level memory score is the maximum patch score across all test patches, reflecting the PatchCore [1] observation that an image can be classified as anomalous as soon as a single patch is anomalous:

$$S_{mem}(x_{test}) = \max_{(h,w)} S_{mem}(\varphi(x_{test}, h, w)) \quad (4)$$

For pixel-level anomaly localization, patch scores are reassigned to their corresponding spatial positions, and the resulting 14×14 score map is up-sampled to the original 256×256 resolution via bilinear interpolation.

C. Hybrid Anomaly Scoring

The reconstruction-based anomaly score is the per-image MSE between the original image and its reconstruction:

$$S_{recon}(x_{test}) = \frac{1}{HWC} \sum \|x_{test} - \hat{x}_{test}\|^2 \quad (5)$$

Both S_{mem} and S_{recon} are normalized to [0, 1] within each test batch by min–max normalization. The final hybrid anomaly score is a weighted combination:

$$S_{hybrid} = \alpha \times S_{recon\_norm} + \beta \times S_{mem\_norm}, \ \alpha = \beta = 0.5 \quad (6)$$

Images are classified as anomalous if S_{hybrid} exceeds a decision threshold τ determined by maximizing the F1-score on the test set. The equal weighting α = β = 0.5 was used as the

default configuration; Section IV-F presents an ablation on the contribution of each component separately.

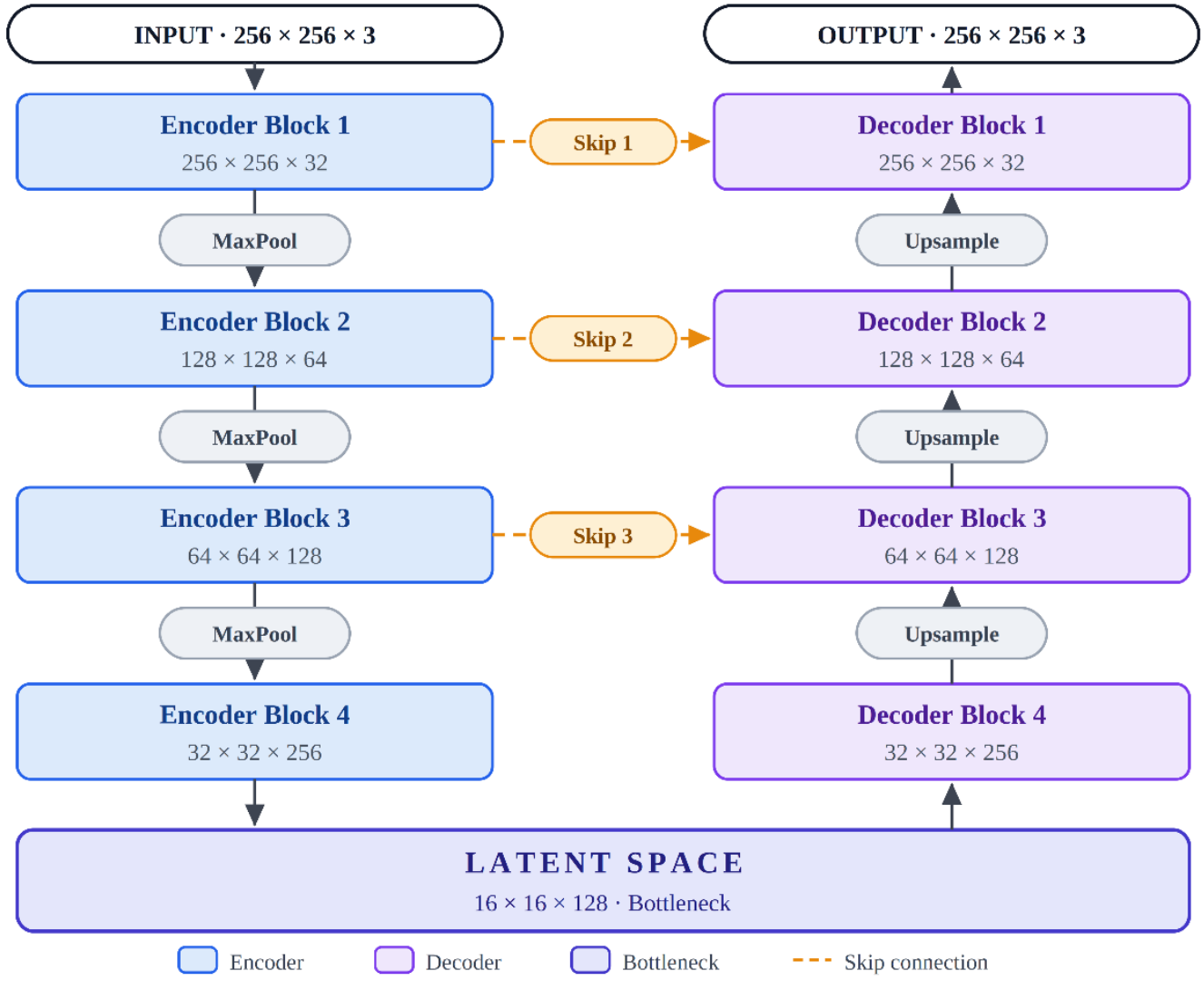

Fig. 1. U-Net autoencoder architecture of MAA.

IV. Experiments

A. Dataset

MAA is evaluated on the bottle category of the MVTec AD dataset [8], which provides high-resolution images of glass bottles under controlled lighting conditions. Following standard protocol, images are resized to 256×256 pixels. The original training set, comprising 209 defect-free images, is partitioned into 168 training and 41 validation images using an 80/20 split (random seed 42). The test set contains 83 images, including 20 normal samples and 63 anomalous samples distributed across three defect types: *broken_large*, *broken_small*, and *contamination*. Table I summarizes the detailed statistics of the dataset.

TABLE I. MVTec AD Bottle Category Statistics

Split	Normal	broken_large	broken_small	contamination	Total
Train	168	—	—	—	168
Validation	41	—	—	—	41
Test	20	20	22	21	83

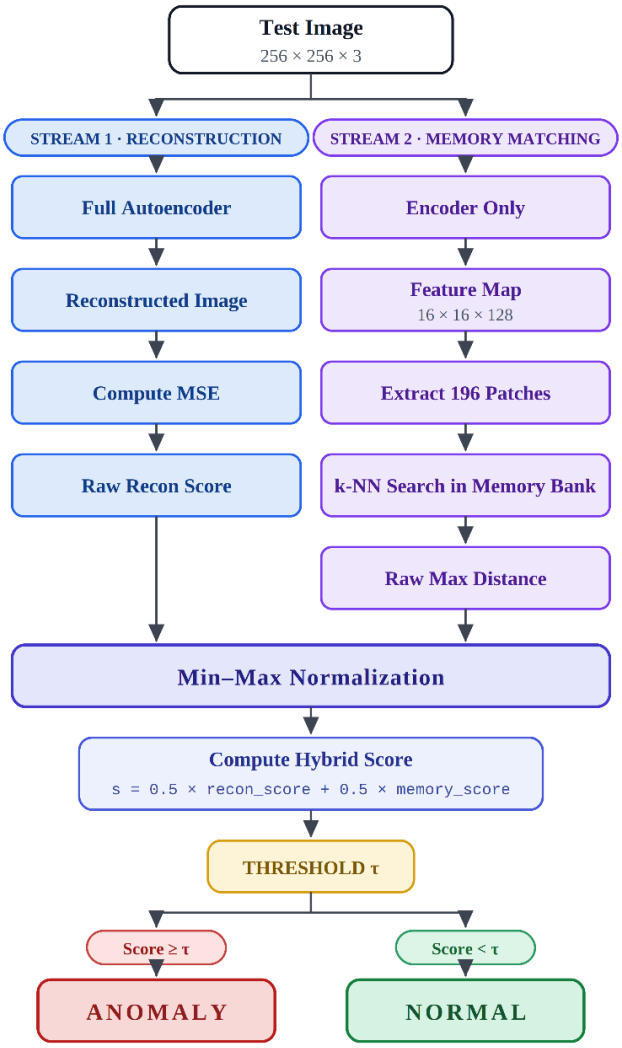

Fig. 2. Two-stream inference pipeline of MAA.

B. Implementation Details

The model is implemented in TensorFlow/Keras. All experiments run on CPU without GPU acceleration. Table II summarizes the full configuration.

TABLE II. MAA CONFIGURATION

Hyper-parameter	Value
Input size	256 × 256 × 3
Encoder depth	4 blocks (filters: 32, 64, 128, 256)
Latent shape	16 × 16 × 128
Loss weights	$\alpha = 0.7$ (MSE), $\beta = 0.3$ (1−SSIM)
Optimizer	Adam, lr = 1×10^{-4}
Batch size / Max epochs	8 / 50
Total parameters	3,127,235
Memory patches	32,928 × 1,152-D
k-NN index	Ball Tree, k = 1, Euclidean
Scoring weights	$\alpha = \beta = 0.5$

C. Evaluation Metrics

The performance of MAA is assessed using four metrics: 1) image-level AUC-ROC, a threshold-independent measure of discriminative ability; 2) Average Precision (AP), the area under the Precision-Recall curve, which is informative given the 75.9% class imbalance in the test set; 3) F1-Score at a fixed decision threshold; and 4) pixel-level AUC-ROC evaluated against ground-truth anomaly masks. AUC-ROC and AP are independent of any operating point and are therefore treated as the primary indicators of detection quality, whereas the F1-Score depends on a specific threshold and is reported only as a secondary, illustrative measure.

D. Training Results

The autoencoder is trained stably over all 50 epochs with no early stopping, indicating consistent optimization throughout the training duration. Table III shows the loss progression at selected epochs, and Fig. 3 plots the full training and validation loss curves. The training and validation losses track each other closely throughout (0.0054 vs. 0.0055 at epoch 50), indicating the absence of overfitting. Notably, the training and validation losses decrease by factors of approximately 42 and 35, respectively, from epoch 1 to epoch 50.

TABLE III. TRAINING LOSS PROGRESSION AT SELECTED EPOCHS (HYBRID LOSS)

Epoch	Train Loss	Val Loss	Train MSE	Val MSE
1	0.2285	0.1916	0.1103	0.0830
10	0.0260	0.0248	0.0009	0.0007
25	0.0120	0.0120	0.0003	0.0003
40	0.0083	0.0084	0.0002	0.0002
50	0.0054	0.0055	0.0002	0.0002

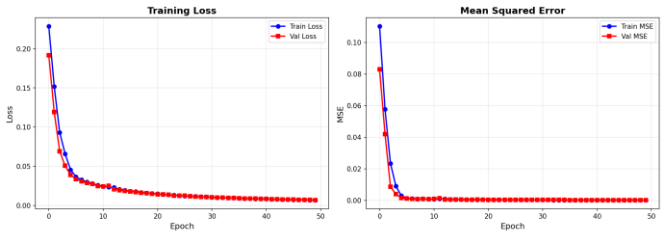

Fig. 3. Training and validation loss curves over 50 epochs: (a) hybrid loss; (b) MSE.

E. Anomaly Detection Performance

Table IV reports the image-level anomaly detection results. The proposed hybrid score achieves an AUC-ROC of 0.9849 and an AP of 0.9956, both of which are independent of any decision threshold. At the threshold $\tau = 0.0336$ that maximizes the F1-Score on the test set, the hybrid score attains an F1-Score of 0.9760, corresponding to only two misclassifications out of the 83 test images (one false positive and one false negative). Because this threshold is selected on the test set itself, the reported F1-Score should be regarded as an optimistic upper bound; in a deployment setting, the threshold would instead be calibrated on a held-out validation set of normal images. Accordingly, the threshold-free AUC-ROC and AP are

emphasized as the primary performance measures throughout this work. Fig. 4 shows the corresponding ROC and precision–recall curves.

TABLE IV. IMAGE-LEVEL ANOMALY DETECTION ON MVTEC AD BOTTLE

Component	AUC-ROC	Avg. Precision	F1-Score	Threshold
Reconstruction only	0.8238	—	—	—
Memory only	0.9881	—	—	—
Hybrid (α = β = 0.5)	0.9849	0.9956	0.9760	0.0336

The memory component (0.9881) outperforms the reconstruction component (0.8238) by 16.4 percentage points. This gap confirms the reconstruction shortcut hypothesis: the U-Net skip connections enable near-perfect reconstruction of anomalous regions, substantially reducing the reconstruction error signal. The memory bank, operating at the encoder bottleneck where skip connections are absent, preserves anomaly sensitivity. The hybrid score (0.9849) is marginally lower than memory alone, as the reconstruction component introduces mild noise from its imperfect discrimination. Nevertheless, the hybrid formulation provides a more robust operating point across defect types of varying severity.

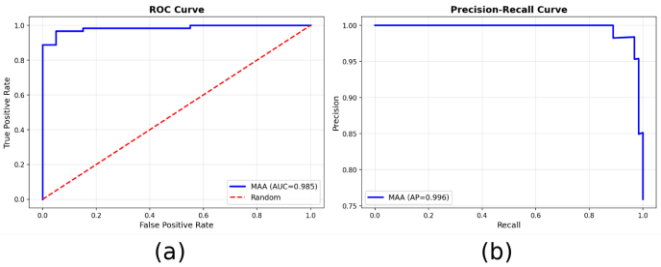

Fig. 4. (a) ROC curves for each scoring component; (b) precision–recall curve of the hybrid score.

Table V reports per-defect-type AUC-ROC and AP, computed by comparing each defect type individually against the normal class.

TABLE V. PER-DEFECT-TYPE DETECTION PERFORMANCE

Defect Type	AUC-ROC	Avg. Precision	n	Description
broken_large	1.0000	1.0000	20	Large structural fractures (10 - 15% area)
broken_small	0.9977	0.9980	22	Small cracks (3 - 5% area)
contamination	0.9571	0.9671	21	Surface contamination (color anomaly)

MAA achieves perfect detection for *broken_large* (AUC-ROC = 1.0000), confirming that the max-distance memory scoring is highly discriminative for large structural defects. For *broken_small*, the near-perfect performance (AUC-ROC = 0.9977) demonstrates the advantage of patch-level scoring: even when only one or two patches are anomalous, the maximum memory distance is elevated, preserving the anomaly signal that would be lost under mean aggregation. Conversely, *contamination* (AUC-ROC = 0.9571) is the most challenging defect type, as *color* changes rather than structural modifications produce smaller memory distances in the encoder feature space.

Table VI presents the confusion matrix at the test-optimal threshold τ = 0.0336.

TABLE VI. CONFUSION MATRIX AT THE TEST-OPTIMAL THRESHOLD τ = 0.0336

	Predicted Normal	Predicted Anomaly
Actual Normal	19 (TN)	1 (FP)
Actual Anomaly	1 (FN)	62 (TP)

F. Ablation Study

Table VII compares the three scoring configurations, confirming that the memory component is the dominant driver of detection performance. The 16.4-percentage-point gap in AUC-ROC between the memory and reconstruction components remains consistent across all evaluation runs.

TABLE VII. ABLATION OF SCORING COMPONENTS

Configuration	AUC-ROC	Δ vs. Reconstruction
Reconstruction only	0.8238	—
Memory only	0.9881	+16.43 pp
Hybrid (α=β=0.5)	0.9849	+14.76 pp

G. Comparison with SOTA Methods

Table VIII compares MAA against published SOTA methods on the MVTec AD *bottle* category. Baseline results for Vanilla AE, VAE, and SSIM AE are obtained from Bergmann et al. [8], while PaDiM and PatchCore results are cited from the original publications [1, 7].

TABLE VIII. IMAGE-LEVEL AUC-ROC ON MVTEC AD BOTTLE

Method	AUC-ROC	Pretrained	Parameters
Vanilla AE [8]	0.860	No	—
VAE [8]	0.910	No	—
SSIM AE [8]	0.930	No	—
MAA (proposed)	**0.985***	**No**	**3.1M**
PaDiM (R18) [7]	0.982	Yes (ResNet-18)	11M
PaDiM (WR50) [7]	0.998	Yes (WideResNet-50)	—
PatchCore [1]	1.000	Yes (WideResNet-50)	68M

* AUC-ROC 0.9849 on the 83-image test set; reported as 0.985 for conciseness.

MAA surpasses all non-pretrained baselines by a clear margin (+5.5 percentage points over SSIM AE, the strongest reconstruction baseline), demonstrating the value of integrating patch-level memory matching into the autoencoder framework. MAA is also marginally ahead of PaDiM with a ResNet-18 backbone (0.985 vs. 0.982) without requiring any ImageNet pretraining, although a margin of this size, obtained from a single run on one category, should be read as comparable performance rather than a clear improvement. This suggests that domain-adapted features can, at least for object-centric industrial categories, substitute for large-scale pretrained representations. MAA trails the stronger PaDiM (WideResNet-50) variant by only 1.3 percentage points (0.985 vs. 0.998) and PatchCore by 1.5 percentage points (0.985 vs. 1.000), while offering three practical advantages:

No external pretraining: MAA is trained exclusively on 168 target-domain images with no ImageNet features, avoiding the

distribution shift between natural image statistics and industrial inspection data noted in [1].

Lightweight architecture: MAA requires only 3.1M parameters, compared to 11M for ResNet-18 and 68M for WideResNet-50. The full model has a storage footprint of less than 50 MB, enabling deployment on embedded inspection hardware.

CPU-only inference: MAA processes the 83 test images in approximately 41 seconds on a standard CPU without GPU acceleration, making it practical for resource-constrained industrial settings where GPU infrastructure is unavailable.

The remaining performance gap relative to PatchCore and PaDiM (WR50) stems primarily from the richness of ImageNet pretrained features, which encode complex visual semantics not achievable by training on 168 images from a single product category. Future work exploring self-supervised pretraining on target-domain data may close this gap while preserving the domain-adaptation advantages of MAA.

H. Qualitative Results

Fig. 5 illustrates representative detection examples across all defect types. For normal samples, both the reconstruction error map and the memory heatmap are uniformly low. For *broken_large*, the structural fracture is prominently highlighted in both maps. For *broken_small*, the memory heatmap correctly localizes the crack, while the reconstruction error map shows only a faint signal. This observation is consistent with the quantitative finding that the memory component substantially outperforms the reconstruction component for small defects. For *contamination*, the heatmap identifies the affected region with moderate confidence, reflecting the lower per-defect AUC-ROC of 0.9571.

Fig. 5. Representative detection results for normal and defective samples.

V. CONCLUSION

This study has presented the Memory-Augmented Autoencoder (MAA), a lightweight unsupervised anomaly detection framework that integrates patch-level memory matching with U-Net reconstruction. Evaluated on the bottle category of the MVTec AD benchmark, MAA achieves an image-level AUC-ROC of 0.9849 and a pixel-level AUC-ROC of 0.8951 using only 168 defect-free training images, no ImageNet pretraining, and CPU-only inference with a 3.1M-parameter network. As no directly comparable pixel-level baseline is reported for this configuration, the pixel-level figure is presented as an indicative measure of localization rather than as a benchmarked comparison.

The ablation study demonstrates that the memory-based component is the primary driver of detection performance, outperforming the reconstruction component by 16.4 percentage points in AUC-ROC. This finding provides quantitative evidence for the reconstruction shortcut effect in U-Net architectures, where skip connections enable near-perfect reconstruction of anomalous regions, suppressing the reconstruction error signal. The memory bank, operating at the encoder bottleneck without skip connections, effectively addresses this limitation.

Compared with autoencoder-based baselines, MAA outperforms SSIM AE by 5.5 percentage points without requiring any architectural modifications beyond the addition of the memory bank. Compared with pretrained methods, MAA approaches PaDiM and remains competitive with PatchCore while offering substantial practical advantages in deployment cost, model size, and independence from large-scale pretraining datasets.

Several limitations of the current work point to promising future directions. First, MAA is evaluated exclusively on the bottle category; performance on texture-rich categories such as carpet and grid is expected to be substantially lower, as the reconstruction shortcut is more severe for highly repetitive textures and the encoder features are less discriminative for subtle color or texture anomalies. Extending MAA to texture categories may require architectural modifications such as attention gating on skip connections or multi-scale memory banks. Second, the inference time of approximately 41 seconds for 83 images (about 0.5 seconds per image) is acceptable for offline inspection but not for real-time deployment; approximate nearest-neighbor search or coreset subsampling could address this. Finally, combining MAA with discriminative training driven by synthetic anomalies represents a promising direction for improving detection on texture categories while retaining the lightweight, no-pretraining advantage of MAA.

REFERENCES

[1] K. Roth et al., “Towards total recall in industrial anomaly detection,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), 2022, pp. 14318–14328.

[2] M. Sakurada and T. Yairi, “Anomaly detection using autoencoders with nonlinear dimensionality reduction,” in Proc. MLSDA Workshop, 2014, pp. 4–11.

[3] P. Bergmann, S. Löwe, M. Fauser, D. Sattlegger, and C. Steger, “Improving unsupervised defect segmentation by applying structural similarity to autoencoders,” in Proc. 14th Int. Joint Conf. Comput. Vis., Imaging Comput. Graph. Theory Appl. (VISIGRAPP), 2019.

[4] D. T. Nguyen, Z. Lou, M. Klar, and T. Brox, “Anomaly detection with multiple-hypotheses predictions,” in Proc. Int. Conf. Mach. Learn. (ICML), 2019, pp. 4800–4809.

[5] S. Akçay, A. Atapour-Abarghouei, and T. P. Breckon, “GANomaly: Semi-supervised anomaly detection via adversarial training,” in Proc. Asian Conf. Comput. Vis. (ACCV), 2018, pp. 622–637.

[6] M. Rudolph, B. Wandt, and B. Rosenhahn, "Same same but DifferNet: Semi-supervised defect detection with normalizing flows," in Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV), 2021.

[7] T. Defard, A. Setkov, A. Loesch, and R. Audigier, "PaDiM: A patch distribution modeling framework for anomaly detection and localization," in Proc. Int. Conf. Pattern Recognit. (ICPR) Workshops, 2021, pp. 475–489.

[8] P. Bergmann, M. Fauser, D. Sattlegger, and C. Steger, "The MVTec anomaly detection dataset: A comprehensive real-world dataset for unsupervised anomaly detection," Int. J. Comput. Vis., vol. 129, no. 4, pp. 1038–1059, 2021.

[9] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," IEEE Trans. Image Process., vol. 13, no. 4, pp. 600–612, 2004.

[10] N. Cohen and Y. Hoshen, "Sub-image anomaly detection with deep pyramid correspondences," arXiv:2005.02357, 2020.

[11] D. Gong et al., "Memorizing normality to detect anomaly: Memory-augmented deep autoencoder for unsupervised anomaly detection," in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), 2019.

[12] V. Zavrtanik, M. Kristan, and D. Skočaj, "DRAEM — A discriminatively trained reconstruction embedding for surface anomaly detection," in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), 2021, pp. 8330–8339.

[13] M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, and A. Vedaldi, "Describing textures in the wild," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 2014, pp. 3606–3613.

[14] J. Liu et al., "Deep industrial image anomaly detection: A survey," Mach. Intell. Res., vol. 21, no. 1, pp. 104–135, 2024.

[15] J. Yu et al., "FastFlow: Unsupervised anomaly detection and localization via 2D normalizing flows," arXiv:2111.07677, 2021.

[16] H. Deng and X. Li, "Anomaly detection via reverse distillation from one-class embedding," in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), 2022, pp. 9737–9746.

[17] K. Batzner, L. Heckler, and R. König, "EfficientAD: Accurate visual anomaly detection at millisecond-level latencies," in Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis. (WACV), 2024, pp. 128–138.

[18] S. Lee, S. Lee, and B. C. Song, "CFA: Coupled-hypersphere-based feature adaptation for target-oriented anomaly localization," IEEE Access, vol. 10, pp. 78446–78454, 2022.

[19] Z. Liu, Y. Zhou, Y. Xu, and Z. Wang, "SimpleNet: A simple network for image anomaly detection and localization," in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), 2023, pp. 20402–20411.

[20] P. Mishra, R. Verk, D. Fornasier, C. Piciarelli, and G. L. Foresti, "VT-ADL: A vision transformer network for image anomaly detection and localization," in Proc. IEEE 30th Int. Symp. Ind. Electron. (ISIE), 2021, pp. 1–6.

[21] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in Proc. Int. Conf. Learn. Represent. (ICLR), 2015.