Supplementary Material: An Incremental Unified Framework for Small Defect Inspection

Jiaqi Tang[1,2,3], Hao Lu[1,2], Xiaogang Xu[4], Ruizheng Wu[5], Sixing Hu[5], Tong Zhang[6], Tsz Wa Cheng[6], Ming Ge[7], Ying-Cong Chen[1,2,3⋆], and Fugee Tsung[1,2]

[1] Hong Kong University of Science and Technology (Guangzhou)
[2] Hong Kong University of Science and Technology
[3] HKUST(GZ) – SmartMore Joint Lab
[4] Chinese University of Hong Kong
[5] SmartMore Corporation
[6] Hong Kong Industrial Artificial Intelligence & Robotics Centre
[7] Hong Kong Productivity Council
{jtang092, hlu585}@connect.hkust-gz.edu.cn
xiaogangxu00@gmail.com {ruizheng.wu, david.hu}@smartmore.com
{jacquelinezhang, alancheng}@hkflair.org mingge@hkpc.org
yingcongchen@hkust-gz.edu.cn season@ust.hk

Abstract. In this supplementary material, we first extend the related work from the main paper. Then, we provide an in-depth explanation of our proposed network architecture, encompassing both the Reconstruction Network and the Semantic-Discriminative Network. We also present the pseudo-code for our novel updating strategy. Besides, we validate the semantics spaces to support our assumption. Additionally, we examine a failure case and analyze the reason for negative forgetting measurement (FM), memory size, efficiency, and limitations of our current approach. This analysis leads us to propose potential directions for future research, aiming to advance the field further.

1 Related Work (continued)

General Incremental Learning Incremental learning (also known as continuous learning, lifelong learning) aims to enable AI systems to continuously acquire, update, accumulate, and utilize knowledge in response to external changes [14]. The primary challenge in incremental learning is to alleviate catastrophic forgetting, and numerous studies have been conducted to design methods targeting this objective [14]. Typically, continuous learning approaches can be categorized into regularization-based methods [1, 7, 16], rehearsal-based methods [3, 9], and dynamic architecture methods [12], etc. Generally, these methods are closely related to their respective downstream tasks [14].

⋆ Corresponding author.

However, unlike other downstream tasks, our continuous learning requires building upon an image reconstruction network. Therefore, we analyze the fundamental cause of catastrophic forgetting in reconstruction networks as feature conflict and propose a suitable method for reducing catastrophic forgetting in reconstruction networks by employing network space operations.

2 Network Implementation

Our framework is composed of two critical learning-based modules: the Reconstruction Network and the Semantic-Discriminative Network. The subsequent sections will delineate the intricacies of each component in detail.

2.1 Reconstruction Network

Following the previous studies [10, 15], we adopt the transformer-based autoencoder for reconstruction. Specifically, we use a four-layer encoder and a four-layer decoder. Table 2 shows the details of this network.

2.2 Semantic-Discriminative Network

The Semantic-Discriminative Network serves as a plug-and-play module designed to enhance the learning of semantic boundaries. It's seamlessly integrated with the Object-Aware Self-Attention (OASA) mechanism. To achieve this, we leverage a transformer-based architecture for the discriminator, focusing on the effective classification of labels. The network is structured around a four-layer transformer, complemented by two fully connected layers, as detailed in Table 3.

2.3 Objective of C_{O_n} and Decision of T.

C_{O_n} is the intermediate feature of the classifier, representing the semantic spaces of different objects. We formulate satisfied C_{O_n} with T feature layers to capture both coarse semantics in shallow layers, and fine-grained semantics from deep layers. The location of these T layers is identical in the discriminator and the reconstruction network since they have the same encoder structure.

3 Experiment Validation of Objects' Semantics Spaces

Fig. 1 shows an example of the feature spaces of various objects in our network. Different objects have varying activations in distinct channels, which reflects their occupied different semantic spaces. Certainly, shared semantic spaces may also happen (like grid and screw).

4 Pseudo-Code of Our Updating Strategy

Algorithm 1 shows our update strategy. This strategy makes two modifications to Vanilla Gradient Descent. The first is retaining prior semantic information by model regulation in updating, and the second is decreasing the rewriting of prior semantics by semantic importance. Both of them make the updated space to minimize the impact on the old model memory, thus avoiding catastrophic forgetting.

5 Selection of New or Old Objects

We verified the feature distribution of the dataset and found they are separated (as Fig. 5). So, the objects' order is not strict. We follow the previous research (CAD [8]), sorting all objects in alphabetical order.

6 Quantitative Evaluation Throughout the Procedure

We quantitatively evaluate the efficacy of our proposed method by examining its performance across the entire task stream. The evaluation focuses on the accuracy of our method and its ability to retain learned information over time, which is paramount in the context of incremental learning.

As depicted in Fig. 3 and Fig. 4, our method consistently outperforms other baselines under most task protocols. Although there are instances where the model's performance in certain intermediate steps is suboptimal, it still maintains the SOTA overall performance in the whole task stream.

7 Explanation of Negative FM.

The reason is that when the AUROC of the previous tasks is lower, the new updating object may improve the overall AUROC, resulting in a negative FM (as Fig. 2).

Since the base AUROC of our method is much higher than other baselines, the performance of the newly integrated object is still higher (although it may cause a decrease overall), leading to a high forgetting measure. Therefore, FM does not reflect the real performance in retaining previous objects or integrating new objects facing different previous AUROC.

8 Memory Size Analysis

First, as Table 1, our network's base backbone is not heavy and takes up less memory space than others. For updating new objects, our saved feature has already been compressed in the spatial dimensions and is much smaller than other baselines, $L \times 1 (\approx 5) << H \times W (\approx 48 \times 48)$.

Table 1: Memory analysis of different frameworks.

	CAD	UCAD	Ours
Fixed (Base Model) ↓	ViT (∼330 MB)	SAM (∼2.4G) + ViT (∼330MB)	Recon. + Classifi. (∼160 MB)
After Updating ↓	$N \times C \times H \times W$	$N \times C \times H \times W$	$L \times N \times C \times \mathbf{1}$ (L is number of layers)

9 Preliminary Analysis in Efficiency

Compared to UniAD [15], while our method has an efficiency loss due to the classification network, it still has high performance (about 5.88 FPS), and it is higher than other SOTA baselines such as Padim [4] (about 1.14 FPS), PatchCore [11] (about 1.28 FPS), Reverse Distillation [5] (about 0.10 FPS), and Dream [6] (about 0.45 FPS). *The problem of efficiency is not the main concern of our paper, but it is a good direction to explore in the future.*

10 Failure Analysis

Despite our algorithm's advancements in mitigating catastrophic forgetting, it encounters challenges with complex structured objects, as illustrated in Fig. 6. While it significantly reduces semantic feature conflicts, outperforming UniAD [15] in handling a broad spectrum of semantic inconsistencies, limitations inherent to the semantic space and network capacity prevent it from capturing all correct semantic information accurately.

11 Limitations

Firstly, our approach does not explicitly represent the semantic space capacity, which is unfavorable for measuring the network ability for object-incremental learning. Besides, feature sharing is also significant in incremental learning since feature sharing on the reconstruction network can reduce the space occupation of individual objects. These limitations are valuable and should be further discussed.

12 Future Work

Given the limitations discussed in Sec. 10 and Sec. 11, a critical future research direction is the explicit evaluation of the semantic space within the reconstruction network. An accurate estimation of the semantic space would enable us to gauge the network's capacity directly. This insight could then be leveraged to dynamically compress or expand the network's space as required, optimizing performance and adaptability.

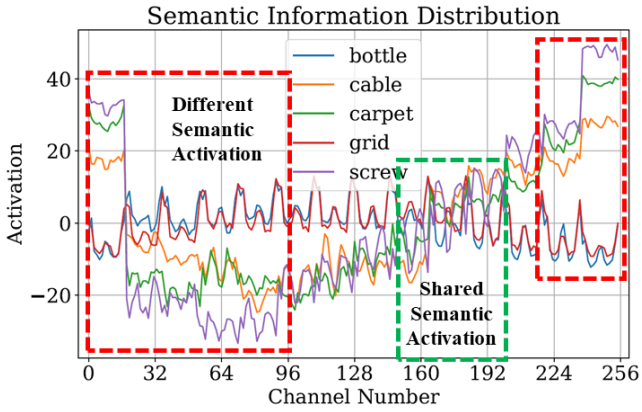

Fig. 1: Example of Objects' Semantic Space.

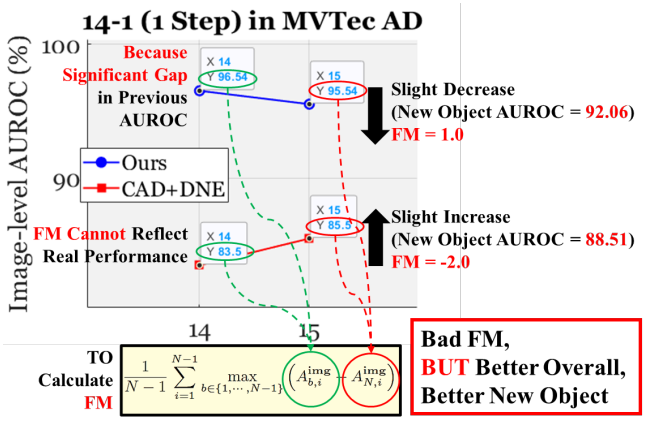

Fig. 2: Explanation of negative FM.

Table 2: Architecture of transformer-based autoencoder with feature extracting network (EfficientNet [13]). Each encoder consists of an Object-Aware Self-Attention (OASA) followed by a feed-forward neural network (FFN), as $Y = \text{FFN}(\text{OASA}(X_Q, X_K, X_V) + X) + X$. Each encoder consists of a vanilla Self-Attention (SA) followed by a feed-forward neural network (FFN), as $Y = \text{FFN}(\text{SA}(X_Q, X_K, X_V) + X) + X$

Layer Type	Position	Activation	Heads	KQV Dim	Feed-forward Dim	Output Size
Input Image	-	-	-	-	-	$H \times W \times 3$
EfficientNet	-	-	-	-	-	$\frac{H}{16} \times \frac{W}{16} \times C$
Embedding Layer	-	-	-	-	-	$\frac{H}{16} \times \frac{W}{16} \times C$
Encoder Layers						
OASA	1	ReLU	2	1024	-	$\frac{H}{16} \times \frac{W}{16} \times C$
Feed-forward	1	ReLU	-	-	1024	$\frac{H}{16} \times \frac{W}{16} \times C$
OASA	2	ReLU	4	1024	-	$\frac{H}{16} \times \frac{W}{16} \times C$
Feed-forward	2	ReLU	-	-	1024	$\frac{H}{16} \times \frac{W}{16} \times C$
OASA	3	ReLU	4	1024	-	$\frac{H}{16} \times \frac{W}{16} \times C$
Feed-forward	3	ReLU	-	-	1024	$\frac{H}{16} \times \frac{W}{16} \times C$
OASA	4	ReLU	8	1024	-	$\frac{H}{16} \times \frac{W}{16} \times C$
Feed-forward	4	ReLU	-	-	1024	$\frac{H}{16} \times \frac{W}{16} \times C$
Decoder Layers						
Self-Attention	1	ReLU	2	1024	-	$\frac{H}{16} \times \frac{W}{16} \times C$
Feed-forward	1	ReLU	-	-	1024	$\frac{H}{16} \times \frac{W}{16} \times C$
Self-Attention	2	ReLU	4	1024	-	$\frac{H}{16} \times \frac{W}{16} \times C$
Feed-forward	2	ReLU	-	-	1024	$\frac{H}{16} \times \frac{W}{16} \times C$
Self-Attention	3	ReLU	4	1024	-	$\frac{H}{16} \times \frac{W}{16} \times C$
Feed-forward	3	ReLU	-	-	1024	$\frac{H}{16} \times \frac{W}{16} \times C$
Self-Attention	4	ReLU	8	1024	-	$\frac{H}{16} \times \frac{W}{16} \times C$
Feed-forward	4	ReLU	-	-	1024	$\frac{H}{16} \times \frac{W}{16} \times C$
Output Layer	-	Sigmoid	-	-	-	$\frac{H}{16} \times \frac{W}{16} \times C$

Table 3: Architecture of the Semantic-Discriminative Network with one transformer backbone and two full connect layers for classifying.

Layer Type	Position	Activation	Heads	KQV Dim	Feed-forward Dim	Output Size
Input Feature	-	-	-	-	-	$\frac{H}{16} \times \frac{W}{16} \times C$
Transformer Backbone						
Self-Attention	1	ReLU	2	1024	-	$\frac{H}{16} \times \frac{W}{16} \times C$
Feed-forward	1	ReLU	-	-	1024	$\frac{H}{16} \times \frac{W}{16} \times C$
Self-Attention	2	ReLU	4	1024	-	$\frac{H}{16} \times \frac{W}{16} \times C$
Feed-forward	2	ReLU	-	-	1024	$\frac{H}{16} \times \frac{W}{16} \times C$
Self-Attention	3	ReLU	4	1024	-	$\frac{H}{16} \times \frac{W}{16} \times C$
Feed-forward	3	ReLU	-	-	1024	$\frac{H}{16} \times \frac{W}{16} \times C$
Self-Attention	4	ReLU	8	1024	-	$\frac{H}{16} \times \frac{W}{16} \times C$
Feed-forward	4	ReLU	-	-	1024	$\frac{H}{16} \times \frac{W}{16} \times C$
Classification Head						
Global Average Pooling	-	-	-	-	-	C
Fully Connected	-	ReLU	-	-	-	$\frac{C}{2}$
Fully Connected	-	ReLU	-	-	-	15
Output Layer	-	Softmax	-	-	-	15

Algorithm 1 Model Updating with Prior Memory

Require: Model parameters: $\theta = (\theta_1, \theta_2, \ldots, \theta_J) \in \mathbb{R}^{C \times W}$, representing the weights of a neural network model across C channels and W weights.

Require: Learning rate: α, the step size used in the gradient descent optimization.

Require: Old weights: θ_j^{old}, the weights of the model from the previous iteration, used for preserving prior learning.

Require: Regulation hyper-parameter: β, a constant to control the influence of old weights on the update.

Require: Channel eigenspace in old objects: V_{old}^T, representing the eigenvectors of the old weight space, used to project updates.

1: Update weight considering old semantics: $\theta_j^{'} \leftarrow \theta_j + \nabla\theta_j + \beta\theta_j^{old}$, where $\nabla\theta_j$ is the gradient of the loss function w.r.t. θ_j. This step combines the current update with a portion of the old weights to maintain prior knowledge.

2: Project gradient to channel space: $\nabla\Theta_j = V_{old}^T \nabla\theta_j$, projecting the gradient onto the space of old channels.

3: Constrain updates: $\Omega(k, c) = k * \log(c)$, defining a function to modulate the updates based on the channel and some constant k.

4: Update gradient: $\nabla\theta_j^* = (V_{old}^T)^{-1} \Omega(k, n) \odot \nabla\Theta_j$, applying the modulation to the projected gradient.

5: Update model: $\theta_j^{'} \leftarrow \theta_j + \nabla\theta_j^* + \beta\theta_j^{old}$, finalizing the update by combining the modulated gradient with the original and old weights.

6: **return** Updated model parameters $\theta^{'}$

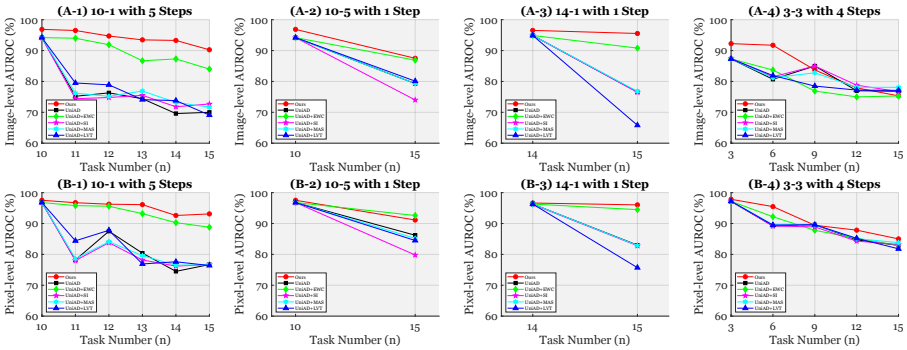

Fig. 3: Quantitative evaluation in all steps (MvTec AD [2]). Here are the accuracy-changing charts in the whole task stream. These charts indicate our method can maintain a low forgetting rate compared with other baselines.

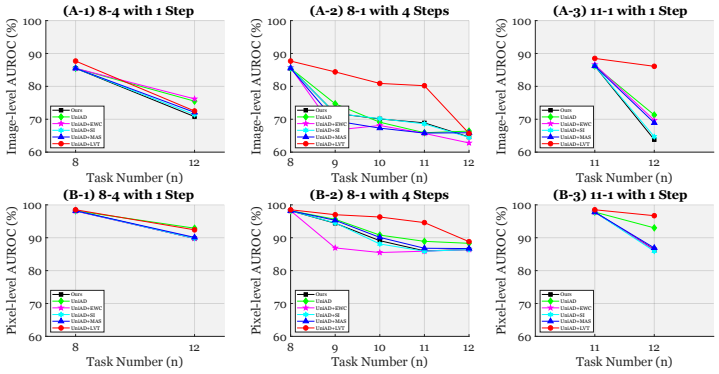

Fig. 4: Quantitative evaluation in all steps (VisA [17]). Here are the accuracy-changing charts in the whole task stream. These charts indicate that our method can maintain a low forgetting rate compared with other baselines.

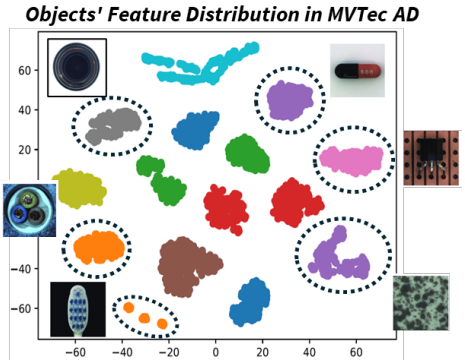

Fig. 5: Explanation of Selecting New/Old Objects.

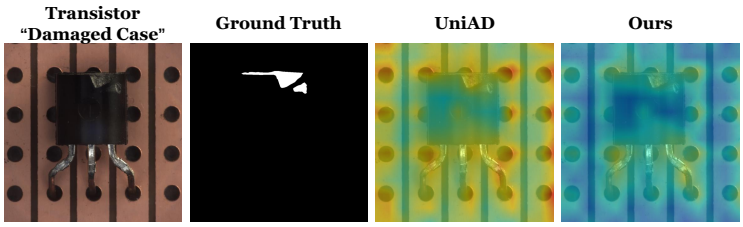

Fig. 6: Failure Case in Transistor. Compared to UniAD [15]. Although our method avoids catastrophic forgetting and reconstructs the corresponding objects as much as possible, it still cannot locate defective regions precisely.

References

1. Aljundi, R., Babiloni, F., Elhoseiny, M., Rohrbach, M., Tuytelaars, T.: Memory aware synapses: Learning what (not) to forget. In: Proceedings of the 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 (CVPR). pp. 9592–9600 (2019)
3. Chaudhry, A., Rohrbach, M., Elhoseiny, M., Ajanthan, T., Dokania, P.K., Torr, P.H., Ranzato, M.: On tiny episodic memories in continual learning. arXiv preprint arXiv:1902.10486 (2019)
4. 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. pp. 475–489. Springer (2021)
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 (CVPR). pp. 9737–9746 (2022)
6. Ezeme, O.M., Mahmoud, Q.H., Azim, A.: Dream: deep recursive attentive model for anomaly detection in kernel events. IEEE Access **7**, 18860–18870 (2019)
7. 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)
8. 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: Proceedings of the 30th ACM International Conference on Multimedia. pp. 2871–2880 (2022)
9. Lopez-Paz, D., Ranzato, M.: Gradient episodic memory for continual learning. Advances in neural information processing systems **30** (2017)
10. Pirnay, J., Chai, K.: Inpainting transformer for anomaly detection. In: International Conference on Image Analysis and Processing. pp. 394–406. Springer (2022)
11. 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)
12. Serra, J., Suris, 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)
13. Tan, M., Le, Q.: EfficientNet: Rethinking model scaling for convolutional neural networks. In: Proceedings of the 36th International Conference on Machine Learning. pp. 6105–6114 (2019)
14. Wang, L., Zhang, X., Su, H., Zhu, J.: A comprehensive survey of continual learning: Theory, method and application. arXiv preprint arXiv:2302.00487 (2023)
15. You, Z., Cui, L., Shen, Y., Yang, K., Lu, X., Zheng, Y., Le, X.: A unified model for multi-class anomaly detection. Advances in Neural Information Processing Systems **35**, 4571–4584 (2022)
16. Zenke, F., Poole, B., Ganguli, S.: Continual learning through synaptic intelligence. In: International conference on machine learning. pp. 3987–3995. PMLR (2017)
17. 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)