The International Journal of Advanced Manufacturing Technology
https://doi.org/10.1007/s00170-026-18080-y

ORIGINAL ARTICLE

# A continual learning framework for defect recognition in additive manufacturing using a progressive online ridge regression approach

**Michele Trovato[1] · Mariorosario Prist[2] · Paolo Cicconi[1] · Andrea Bonci[2] · Geremia Pompei[3] · Lorenzo Longarini[2]**

Received: 16 October 2025 / Accepted: 2 April 2026


**Abstract**

Currently, Additive Manufacturing is revolutionizing the production of complex and customized components across various industries, offering significant advantages in material efficiency, design flexibility, and rapid prototyping. Concurrently, Machine Learning models have become increasingly crucial in Additive Manufacturing, improving decision-making, process efficiency, predictive accuracy, and defect recognition. However, the practical implementation of Machine Learning models in production environments presents significant challenges, such as managing incremental knowledge without causing catastrophic forgetting of old knowledge. The complexity of the additive process and the large number of parameters often require training the model on new data without forgetting previously acquired knowledge. Continual Learning is an emerging practice for incremental knowledge management in Artificial Intelligence. This paper presents a novel Continual Learning approach for class-incremental learning tasks, called Progressive Online Ridge Regression (PORR), based on an extended version of Ridge Regression that fine-tunes a pre-trained Convolutional Neural Network, MobileNetV3. The method is applied to an image analysis problem to recognize defects in Powder Bed Fusion of Polymers. An open-access dataset has been analyzed to validate the approach. The experimental results demonstrate that the proposed approach reduces catastrophic forgetting by optimizing computational resource allocation with respect to accuracy, training time, CPU utilization, and maximum RAM usage.

**Keywords** Additive manufacturing · Powder bed fusion of polymers · Defect recognition · Machine learning · Continual learning

**Nomenclature**

| | |
|---|---|
| AEE | Accuracy after Each Experience |
| AI | Artificial Intelligence |
| AM | Additive Manufacturing |
| BER | Balanced Experience Replay |
| CAD | Computer-Aided Design |
| CIL | Class-Incremental Learning |
| CL | Continual Learning |
| CNN | Convolutional Neural Network |
| DIL | Domain-Incremental Learning |
| DfAM | Design for Additive Manufacturing |
| EWC | Elastic Weight Consolidation |
| FE | Features Extractor |
| FEModel | Features Extractor Model |
| GEM | Gradient Episodic Memory |
| GDumb | Greedy Sampler and Dumb Learner |
| IIL | Instance-Incremental Learning |
| KD | Knowledge Distillation |
| LwF | Learning without Forgetting |
| ME | Material Extrusion |
| MJ | Material Jetting |
| ML | Machine Learning |
| MNV3Large | MobileNetV3 Large |
| MNV3Small | MobileNetV3 Small |
| PBF | Powder Bed Fusion |
| PBF-LB/P | Powder Bed Fusion- Laser Beam of polymers |

✉ Michele Trovato
michele.trovato@uniroma3.it

[1] Dipartimento di Ingegneria Industriale, Elettronica e Meccanica, Università degli Studi Roma Tre, Roma 00146, Italy

[2] Dipartimento di Ingegneria dell'Informazione, Università Politecnica delle Marche, Ancona 60131, Italy

[3] Reparto di Ricerca e Sviluppo, Intelligenza Artificiale, Syncode Scarl, Fermo 63900, Italy

Published online: 20 April 2026

| | |
|---|---|
| PORR | Progressive Online Ridge Regression |
| RR | Ridge Regression |
| Tg | glass transition Temperature |
| Tm | melting temperature |
| TIL | Task-Incremental Learning |

## 1 Introduction

Additive Manufacturing (AM) is highly significant across various industries, including aerospace, automotive, healthcare, consumer goods, and industrial manufacturing, owing to its ability to produce complex, lightweight, customized components layer by layer, with efficient material distribution and design flexibility [1]. In this context, Machine Learning (ML) can support product development by leveraging AM's ability to realize complex topologies and facilitate integrated product-process optimization [2]. ML can detect defects using sensors, predict them, and compensate for them, such as cracks, delamination, distortion, rough surfaces, lack of fusion, porosity, and process instability, in in situ applications [3]. Common ML approaches are Deep Convolutional Neural Networks [4] and Support Vector Machines [5]. These algorithms are often used for classification and recognition in image-based analysis. ML is also used for improving predictive maintenance and condition-based monitoring [6].

Implementing ML in the AM industry can be complex and requires several challenges related to modeling, deployment, and informatics infrastructures. Focusing on training activities, there is substantial variability in parameters related to geometry, materials, and processes. Moreover, the possible irregularities, defects, and faults related to the process level are varied. To reduce complexity and enhance predictive reliability, the training phases can focus on specific product and material classes. However, these training activities typically produce static ML models that can become obsolete as production requirements and boundary conditions evolve. On the other hand, training existing models on new data degrades previously learned tasks, leading to catastrophic forgetting [7]. Adaptive learning models, such as Continual Learning (CL), can make real-time adjustments and overcome the catastrophic forgetting limitation. These models assume that all relevant data are available from the outset and that the appearance of new data does not affect the system. CL methods effectively integrate new tasks while preserving strong performance on previously learned ones. One possible approach to face this challenge is to retrain the model whenever a new condition emerges. However, as data sizes increase, long-term retention of process data becomes impractical due to hardware limitations or corporate policies [8–11].

This study presents an innovative CL approach for classification tasks that addresses key challenges, including maintaining performance and computational efficiency. The proposed approach is based on a modified version of Ridge Regression (RR) [12] that solves it online; therefore, it is called Progressive Online Ridge Regression (PORR). The methodology focuses on fine-tuning the MobileNetV3 model [13], a pre-trained Convolutional Neural Network (CNN). The developed ML model has been trained and tested with an open-access image classification dataset [14, 15] related to Powder Bed Fusion (PBF) of Polymers defects to validate the proposed approach. The effectiveness of the proposed method was evaluated with respect to accuracy, training time, CPU usage, and maximum RAM usage.

The remainder of the paper is organized as follows: Sect. 2 provides a literature review. Section 3 presents the proposed approach. Section 4 shows the test case analyzed to validate the effectiveness of the proposed method. Sections 5 and 6 describe the results and conclusions.

## 2 Literature review

### 2.1 Additive manufacturing and design for additive manufacturing

AM has been identified as a pillar of Industry 4.0 [16] because it embodies the fundamental qualities of digital transformation: connection, adaptability, and integration with Artificial Intelligence (AI). Additive methods enhance personalization and flexibility, enabling on-demand production of highly customized objects without the lengthy, costly retooling required by traditional methods. AM promotes innovation by enabling rapid prototyping and reducing time-to-market for new products [17]. Innovative machines can provide real-time data feedback, aiding maintenance and process improvement. The AM process differs from traditional manufacturing methods by enabling the production of customized and complex geometries, thereby giving designers greater freedom [18]. The process begins with the design of a digital model using Computer-Aided Design (CAD) tools, which define the geometry of the object to be produced. Finite Element Method tools can be integrated into the design workflow to optimize the design and conduct simulations to verify the product's functionality [19]. Once the CAD model is complete, it is converted to a machine-readable format, such as STL, 3MF, or AMF [20]. The resulting data exchange file is processed by the build preparation software, which divides the CAD model into horizontal layers. During this phase, the operator, within the software, sets the process path and determines the parameters such as

layer thickness, internal filling, and laser scan velocity. The resulting file is sent to the production machine.

AM technologies are regulated by ISO/ASTM 52900:2021 [21], which defines seven categories of AM technologies: Binder Jetting, Direct Energy Deposition, Material Extrusion (ME), Material Jetting (MJ), Sheet Lamination, Vat Photopolymerization, and PBF, where thermal energy selectively fuses regions of a powder bed.

Design for Additive Manufacturing (DfAM) is a branch of engineering design that considers all known issues related to the AM process, from the early phases to the quality inspection of the built component [22]. In a specific AM process, using DfAM tools and methods may reduce the risk of failure and defects in the produced parts, thereby enabling easier-to-manufacture components [23].

#### 2.1.1 Powder bed fusion of polymers

PBF-Laser Beam of polymers (PBF-LB/P) is a relevant AM process due to its versatility and widespread adoption in industry. This process produces parts layer by layer by melting powder particles using a high-powered laser. The materials involved in this process are polymers and composites. Some of the materials utilized are PA12, PA11, PA6, TPU, PEEK, and PEKK [24]. This paper analyzes PBF-LB/P process, in which a high-powered laser ($CO_2$, diode, or fiber) selectively scans the preheated polymer powder bed, following the cross-sectional pattern of the object. The laser increases the powder's temperature, providing only the energy required to melt the powder. Laser scanning is controlled by galvanometer mirrors, which quickly and accurately direct the beam. Once the first layer is completed, the build platform lowers by one layer thickness, and a new layer of powder is applied. The process continues layer by layer, with each new layer fusing with the previous one until the part is fully built. The chamber is maintained at a controlled temperature to minimize thermal stress and prevent defects such as curling or warping during fabrication. After the PBF-LB/P process, the chamber can cool down gradually, avoiding internal stress and cracking [25].

Several defects and process-related challenges can affect part quality, mechanical properties, and overall results of the PBF-LB/P process [26]. Process parameters that most strongly influence the overall quality and the final mechanical behavior of the part are layer thickness and scan spacing [27]. The main process parameters and their impact on the process are reported in Table 1.

**Table 1** PBF-LB/P parameters and their impact on the process

| Parameters | Impact on Process | Ref |
|---|---|---|
| Laser Power, Energy Density | Affects densification behaviour, porosity, and layer bonding | [28] |
| Scan Speed, Hatch Spacing | Controls fusion consistency and thermal gradients | [29] |
| Powder Bed Temperature | Prevents warping, shrinkage, and curling | [29] |
| Build Chamber Temperature | Maintains uniform processing conditions | [30] |
| Powder Distribution, Recoating | Ensures uniform layers and prevents defects | [30] |

#### 2.1.2 Machine learning applications in PBF

The geometrical variability of the produced parts requires rapid and reconfigurable tools and methods for defect recognition. Distortion, cracking, lack of fusion, excessive melting, non-uniform powder deposition, porosity, delamination, and other process-induced defects affect PBF-LB/P processes [31].

Nowadays, AI methods, such as ML algorithms, are used to achieve this task by analyzing sensor data. Real-time image analysis of the powder bed surface is one of the most used practices for quality control in this field [32]. Generally, Process Monitoring Level (also called *in situ monitoring*) refers to real-time control of the additive process via sensors.

Supervised Learning algorithms are ML tools frequently used to classify and identify problems and imperfections in additive processes [2]. For example, in [33], a hybrid ML model was trained using data derived from back reflection, visible, infrared, and structure-borne acoustic emission. The recorded signals were classified into a lack of fusion, conduction mode, and keyhole. In [34], signals were captured using a microphone, acoustic signals were used for quality monitoring, and defect identification was achieved using the Deep Belief Network architecture. In [35], a CNN has been used to classify infrared thermography recordings of failures induced by curling and other process irregularities that affect mechanical properties and part quality in polyamide powders. In [36], various deep learning models were trained and tested to establish relationships between acoustic signals and spatter characteristics during the AM phase. A microphone recorded acoustic signals, and spatter information was collected simultaneously by a high-speed coaxial camera.

ML techniques are most used as decision-support tools rather than as autonomous control mechanisms. The most widespread use is real-time anomaly detection, in which monitoring data, such as thermal images or optical signals, are continuously analyzed to identify deviations from nominal process behavior. When such anomalies are detected, the building is typically allowed to continue. At the same time, the system records the event's layer number and spatial location and informs the operator either during or after the manufacturing phase [37]. ML is also used to identify

systematic process drift that develops gradually over the course of a build or across multiple builds. By recognizing trends in monitoring signals that deviate from historical baselines, such as changes in thermal response or signal intensity, the system can indicate emerging issues related to laser stability, optics contamination, or powder aging. In practice, this information is used to schedule maintenance or recalibration before subsequent builds rather than to stop the ongoing process [38, 39].

One limitation of using ML methods in Process Monitoring is the potential issues with model updating. Variations in CAD models, novel defects, and other new conditions may require an additional training phase for the ML model. Training the ML model on novel data often leads to catastrophic forgetting. CL methods can help reduce these risks; however, these methods require fine-tuning strategies to obtain reliable models.

Some commercial machines equipped with integrated monitoring systems are available for PBF-LB/P. These solutions typically incorporate sensors, cameras, and, in some cases, third-party software to facilitate the control workflow for in-situ applications. Currently, ML models are starting to be integrated into commercial machines, leveraging machine vision and sensor data. Tables 2 and 3 provide an overview of some software, tools, and machines that integrate ML and/or control systems in the AM field. The integration of ML into AM machines is an ongoing area of research, necessitating continuous system updates.

**Table 2** Some commercial software that uses ML in the AM field

| Tools and Control Software | AM Process | Functionality | Ref |
|---|---|---|---|
| EOS Smart Fusion | Metal PBF | Component of EOS Smart Monitoring, real-time process control by automatically adjusting laser power during the construction phase. | [40] |
| AMiRIS Standard; AMiRIS-LF; AMiRIS Inside | PBF | Analyze the melt pool and individual layers in real time, capturing large volumes of data to map the manufacturing state layer by layer during building phase. | [41] |
| Octo Nexus AI/OctoEverywhere AI | ME | Open-source web interface for managing and monitoring machines. It allows users to control build jobs, adjust settings, and monitor remotely via a web browser. | [42] |
| Obico/Obico AI Monitoring | ME | Obico AI Monitoring is a smart machine monitoring and control platform with cloud-accessible remote access and AI-driven failure detection. | [43] |
| Markforged Blacksmith | ME | AI-driven software feature designed to bring in-process quality control and adaptive manufacturing to Markforged industrial machines. | [44] |

**Table 3** Some commercial AM machines with integrated machine vision

| Machines | AM Process | Functionality | Ref |
|---|---|---|---|
| 3D Systems – SLS 380 | PBF-LB/P | Closed-loop process controls for laser manufacturing, aimed at maintaining thermal uniformity and quality of manufactured parts. | [45] |
| Formlabs Fuse 1 | PBF-LB/P | Closed-loop control system for laser power, scan speed, and powder bed temperature. | [46] |
| Inkbit Vista | MJ | Closed-loop control combined with machine vision and ML to monitor and correct process errors. | [47] |
| EOS FORMIGA P 110 Velocis | PBF-LB/P | Closed-loop for continuous temperature control by spot pyrometer | [48] |

## 2.2 Continual learning: category and strategies

CL is the process through which ML models are incrementally updated to adapt to new data, evolve conditions, and learn new skills without compromising those already learned [49]. The CL process is classified into four categories [50, 51] and five strategies [52]. The CL categories can be summarized into four groups: Instance-Incremental Learning (IIL), Domain-Incremental Learning (DIL), Task-Incremental Learning (TIL), and Class-Incremental Learning (CIL).

In IIL, an instance refers to a single example or data unit used to train the model. In this approach, the system is trained on new examples of known defects, improving its ability to recognize them. The model receives new data instances, but the classification task remains unchanged. The second category, the DIL, addresses the classification problem in different environments or with variations in operational conditions. The domain refers to the environment or context in which the data is collected and utilized. This approach can include variables such as machine type, material, environmental conditions, or data acquisition method. The model learns to adapt to new situations without forgetting what it has already learned in the original domain, and no new defects are introduced for classification. In the TIL category, the model must solve a specific problem or objective task. The model progressively learns new tasks, each with a different purpose or problem type. In the fourth category, CIL, the model progressively learns to recognize new defects while retaining previously learned classes. For

example [53], proposed a Pseudo Replay-based Class CL for Online New Category Anomaly Detection. CIL methods are suitable for process monitoring in AM. This category of models can initially be trained to classify defects such as cracks and distortions, and then learn to recognize new defects, such as voids or thermal deformations, without forgetting previously learned classes. Despite its complexity in industrial implementation, a CIL method is investigated in this work to support the incremental learning of new knowledge from PBF monitoring of polymers. The advantage of using CIL methods lies in understanding a model each time a new defect appears during use. However, this model still suffers from catastrophic forgetting issues after learning new data.

The literature describes different strategies for mitigating the effects of catastrophic forgetting in CL applications, preserving previously acquired knowledge, and maintaining performance. The main strategies are *Replay-Based*, *Regularization-Based*, *Optimization-Based*, *Representation-Based*, and *Architecture-Based*. These methods are generally applied in neural networks, but not exclusively. The *Replay-Based* approach involves storing a subset of past data or generating pseudo-representations of previous experiences to retrain the model. This technique helps preserve prior knowledge and enhance model robustness to forgetting by periodically revisiting and mixing past and new data during training. Methods in this category can be divided into three sub-directions: experience replay [54], generative replay [55], and feature replay [56]. The *Regularization-Based* approach mitigates catastrophic forgetting by adding penalties to the loss function, discouraging significant updates to relevant parameters for previously learned tasks, and maintaining stability in the neural network's weights. This method involves explicit regularization terms to balance old and new tasks, often requiring a frozen copy of the old model for reference. There are two sub-directions: weight regularization and function regularization. Weight regularization selectively penalizes variation in network parameters, with typical implementations adding a quadratic penalty to the loss function based on each parameter's importance for previous tasks, as seen in [57] and its advanced versions [58]. Function regularization targets the prediction function's intermediate or final output, often employing Knowledge Distillation (KD) [59], where the previously learned model serves as a teacher to the currently trained model. The *Optimization-Based* approach focuses on finding a solution that balances performance on new and old tasks by carefully adjusting learning rates or selectively updating parameters. This approach adapts the model to new data without compromising previously acquired knowledge. This approach involves adding additional terms to the loss function and explicitly designing and manipulating optimization programs. Techniques such as gradient projection are commonly used, in which parameter updates are aligned with the direction of experience replay, for example, as in [60]. The *Representation-Based* approach seeks to learn a stable, shared representation that generalizes across tasks, allowing the model to leverage previously learned features, reduce the risk of forgetting, and facilitate CL. This can be achieved through self-supervised learning with contrastive loss, as seen in [13], which improves robustness by interpolating instances of old and new tasks. Pre-training can benefit downstream CL by enhancing knowledge transfer and robustness to forgetting [61]. The *Architecture-Based* approach dynamically adjusts the neural network architecture to accommodate new tasks by expanding the network or using modular components. This technique aims to isolate changes related to new tasks, preventing interference with previously learned knowledge and supporting structured CL. This method involves implementing task-specific parameters through parameter allocation [62], model decomposition [63], and modular networks [64].

An ML model for defect recognition in PBF-LB/P processes requires both efficacy and efficiency, as well as a CL approach to acquire new knowledge. This paper investigates a *Representation-Based* approach in which a CNN model provides the tools to extract semantic features (pre-trained on a large amount of data), simplifying the task for the fine-tuned output layer. Efficiency is guaranteed by continually training only the last layer rather than the whole model. In particular, the paper considers a modified version of the Ridge Regression (RR) technique [12], a linear regression model that uses a mathematical approach to solve it in an online form. The methodology is applied to fine-tune the pre-trained MobileNetV3 model to reduce the overall computational cost [65].

## 3 Proposed approach

The paper presents a novel CL approach for defect-recognition analysis from images in AM. The objective is to avoid the catastrophic forgetting of traditional ML models and maintain performance. The approach is validated in the context of PBF-LB/P, considering a dataset of images from a powder bed surface (Sect. 4). The approach is based on the CIL method and uses a *representation-based* strategy.

Figure 1 describes the methodology analyzed in this paper, which involves a pre-trained CNN model and an RR model for fine-tuning the network's last layer (output layer). MobileNetV3 models are used as pre-trained CNN models for classification problems in image analysis [65]. Using fine-tuning with an RR model, the final model improves

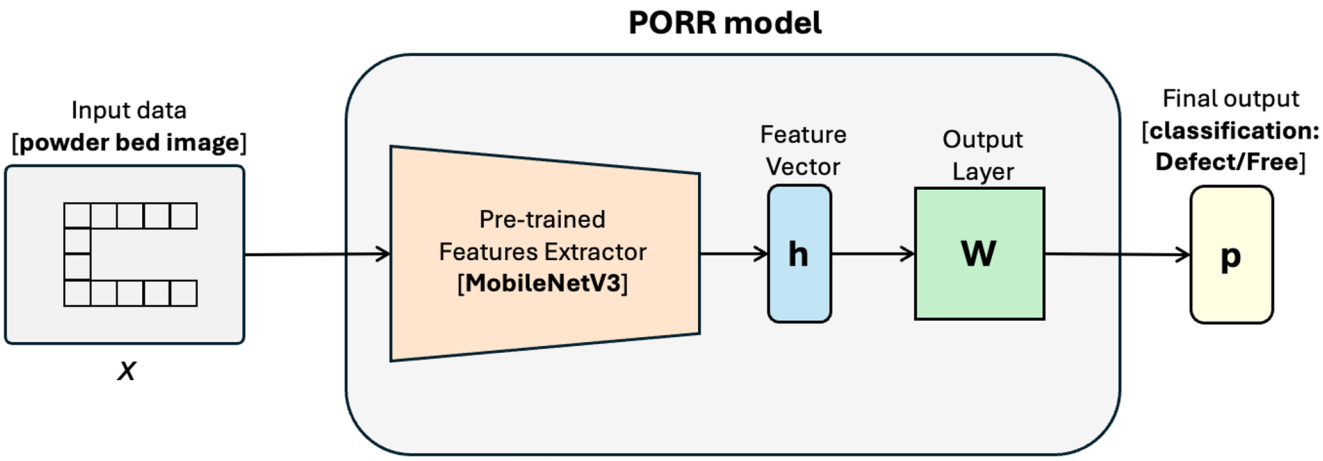


**Fig. 1** The PORR approach for classifying images from the powder bed

its ability to solve the downstream task with a few learning shots rather than starting training from scratch. Therefore, the proposed architecture comprises a frozen and pre-trained Feature Extractor (FE) that embeds the image input data ( $x$) to produce a feature vector ( $h$). This is followed by an output layer ( $W$) that maps the embedding, which can be considered the semantic representation of the input, to the final prediction. Following the feature extraction step, the output $h$ passes through a trainable weight matrix $W$, which adapts the model to the current task. Since the pre-trained model is already available, the output layer is continuously adapted each time new data arrives, obtaining continual fine-tuning. The final output $p$ corresponds to the model's classification result based on the extracted features and the learned parameters $W$. The following subsections describe each part of this architectural design in detail.

### 3.1 Pre-trained feature extractor models

The pre-trained model is a deep learning model already fitted for a general task, such as the downstream task. If the pre-trained model has been trained on a large amount of high-quality data, it converges to strong performance much faster than a randomly initialized model, since its weights are already adapted to capture the data's semantics. The pre-trained model can be defined as a Feature Extractor Model (FEModel) that extracts latent features from the input data. The latent features of the input data are collected into a single vector called an "embedding". For image classification, two advanced CNN architectures were analyzed: MobileNetV3 Large (MNV3Large) and MobileNetV3 Small (MNV3Small) [65]. Both models were used to extract features related to the images' semantic content. MobileNetV3 is a highly efficient CNN architecture with two variants: MobileNetV3-Large and MobileNetV3-Small, each optimized for different computational capacities and accuracy trade-offs. MobileNetV3-Large is designed for tasks requiring higher accuracy with more computational resources. It can handle more complex patterns in the data due to its larger capacity. At the same time, MobileNetV3-Small is optimized for environments where computational resources and power efficiency are critical and offers a balanced trade-off between accuracy and efficiency. MobileNetV3-Large comprises 15 bottleneck layers, one standard convolutional layer, and three pointwise convolutional layers. At the same time, MobileNetV3-Small uses fewer parameters, smaller expansion ratios in the bottlenecks, and less frequent use of the squeeze-and-excitation blocks, making it faster and lighter. To build an efficient model, MobileNetV3 integrates key components from previous architectures, including MobileNetV1, MobileNetV2, and MnasNet. These are combined to form the "bneck" block, the core unit for extracting meaningful features from the input data [66].

This paper evaluates both MNV3Large and MNV3Small as FEModel, and compares the resulting performance in terms of accuracy, training time, CPU Usage, and max RAM usage.

### 3.2 Progressive online ridge regression method

The proposed approach, PORR, is applied to a pre-trained model using a *Representation-based* strategy. A feature extractor processes the input data fixed for each new learning experience, while the output layer is continually adapted. The PORR leverages a mathematical technique to fine-tune an ML model in a CL scenario with exact precision.

**Description** Let $D = \{X, Y\}$ be a dataset where $X \in R^{N_n \times N_x}$ and $Y \in R^{N_n \times N_y}$, with $N_n$ representing the number of examples, $N_x$ the dimensionality of the

input data, and $N_y$ the dimensionality of the output data. Suppose this dataset can be partitioned into multiple experiences $D = D_1, ..., D_E$, where each $D_e = \{X_e, Y_e\}$ for $e = 1, \ldots, E$. If a learning rule (the RR) is applied to the entire dataset $D$ in one pass, or an exact learning rule is applied incrementally to each partition $D_e$, the resulting model will have the same learned weights in both cases. Formally, if $W_D$ is the weight matrix learned by training on $D$, and $W_{D_1, \ldots, D_E}$ is the weight matrix learned incrementally on the partitions $D = D_1, ..., D_E$, then $W_D = W_{D_1, \ldots, D_E}$ [67], which means the model trained on the full dataset $D$ is equivalent, in terms of learned parameters, to the model trained on the incremental experiences $D = D_1, ..., D_E$, ensuring consistency in the CL framework. Specifically, RR is applied to fit a layer on top of a pre-trained feature extractor model, which extracts embeddings from the input data.

**Assumption** The Feature Extractor function can be formalized as:

$$h_i = FE(x_i) \tag{1}$$

where $h_i \in R^{N_h}$ is the embedding of the i-th input data $x_i$ processed by the $FE$ model, and $N_h$ is the embedding dimensionality. The final layer, trained using RR, is represented by the weight matrix $W$ and produces the output through a simple matrix multiplication. This can be formalized as

$$o_i = W \cdot h_i \tag{2}$$

where $o_i \in R^{N_y}$ is the output of the model given by the multiplication of the last layer represented by the matrix $W$ and the embedding $h_i$. The RR is a one-shot learning rule formalized in the following way

$$W = Y^T H \cdot (H^T H + \lambda I)^{-1} \tag{3}$$

where $W \in R^{N_y \times N_h}$ is the weights matrix of the fitted final layer, $Y \in R^{N_n \times N_y}$ is the ground truth of the dataset D, $H \in R^{N_n \times N_h}$ is the embeddings of all input data X of the dataset D, $I$ is the identity matrix and $\lambda$ is the Tikhonov regularization hyperparameter [33]. $\lambda I$ is part of the learning function that can provide a regularization that avoids the model overfitting. Let $A = Y^T H$ and $B = H^T H$, the RR formula can be rewritten as

$$W = A \cdot (B + \lambda I)^{-1} \tag{4}$$

where the $A$ and $B$ matrices are related to the entire dataset $D$. The key to the PORR technique lies in expressing $A$ and $B$ as the sums of $A_e$ and $B_e$ for each experience $e$.

**Definition** Let $R \in R^{N_y \times N_n}$ and $S \in R^{N_n \times N_h}$. The partitions of R and S matrices are conformable for block multiplication if:

- The column partition of R matches the row partition S.
- The width of the block matrix $R_j$ is the same as the height of the block matrix $S_i$, for any $j = 1, \ldots E$ and $i = 1, \ldots E$.

**Theorem (Column-Row partition matrix rule)** *Let $R \in R^{N_y \times N_n}$ and $S \in R^{N_n \times N_h}$, if these two matrices are partitioned in a conformable way into block matrices (sub-matrices) as.*

$$R = [R_1 \ \ldots \ R_E] \text{ and } S = \begin{bmatrix} S_1 \\ \vdots \\ S_E \end{bmatrix} \tag{5}$$

Where $R_j$ is a $N_y \times N_e$ for $e = 1, ..., E$ block matrix with $N_1 + N_2 \ldots + N_E = N_n$ and $S_i$ is a $N_e \times N_h$ for $e = 1, ..., E$ block matrix with $N_1 + N_2 \ldots + N_E = N_n$. Then

$$\begin{aligned} RS &= [R_1 \ \ldots \ R_E] \begin{bmatrix} S_1 \\ \vdots \\ S_E \end{bmatrix} \\ &= R_1 S_1 + R_2 S_2 + \ldots + R_E S_E \\ &= \sum_{e=1}^{E} R_e S_e \end{aligned} \tag{6}$$

**Method** Let $Y \in R^{N_n \times N_y}$ and $H \in R^{N_n \times N_h}$. Let Y and H be partitioned into multiple experiences $e = 1, \ldots, E$, with $N_1 + N_2 \ldots + N_E = N_n$ then.

$$\begin{aligned} Y = \begin{bmatrix} Y_1 \\ \vdots \\ Y_E \end{bmatrix} \Longrightarrow Y^T = \\ [Y_1^T \ \ldots \ Y_E^T] \text{ and } H = \begin{bmatrix} H_1 \\ \vdots \\ H_E \end{bmatrix} \end{aligned} \tag{7}$$

Equation 4 can be rewritten considering:

$$\begin{aligned} A &= \sum_{e=1}^{E} Y_e^T H_e \text{ and} \\ B &= \sum_{e=1}^{E} H_e^T H_e \end{aligned} \tag{8}$$

At the end of the last experience E, W is the same as the one provided by the RR applied to the entire dataset D. Based on previous consideration, the proposed approach involves reworking the RR formula as an exact learning rule within a

CL scenario. The block diagram of PORR is shown in Fig. 2, and the pseudocode for the PORR algorithm is presented in Listing 1.

***Input***: $D = \{Xe, Ye\}$ ***for*** $e = 1, ..., E$
***Output***: W denotes the weights of the output layer

Let $A=0$, $B=0$
***for*** $(X_e, Y_e)$ ***in*** $(X, Y)$:
$H_e = FE(X_e)$
$A_e = Y_e^T H_e$
$B_e = H_e^T H_e$
$A = A + A_e$
$B = B + B_e$
$W = A \cdot (B + \lambda I)^{-1}$
***End***

The PORR pseudocode methodology for updating the readout matrix W for each experience

The choice of ridge regression as the primary update rule is intimately tied to the proposed architecture. Given that the feature extractor is a frozen pre-trained CNN and only the final linear layer is adapted, ridge regression provides a closed-form solution that can be rewritten in terms of sufficient statistics. This enables exact incremental updates equivalent to batch training on the whole dataset, rather than the approximation achievable with iterable methods. Compared with SGD-based online optimization, this method circumvents the need for iterative gradient steps and reduces the number of hyperparameters to tune. Furthermore, unlike replay-based neural methods, PORR does not require storing raw past samples and instead updates only the small readout layer, resulting in a substantial reduction in memory usage and training time.

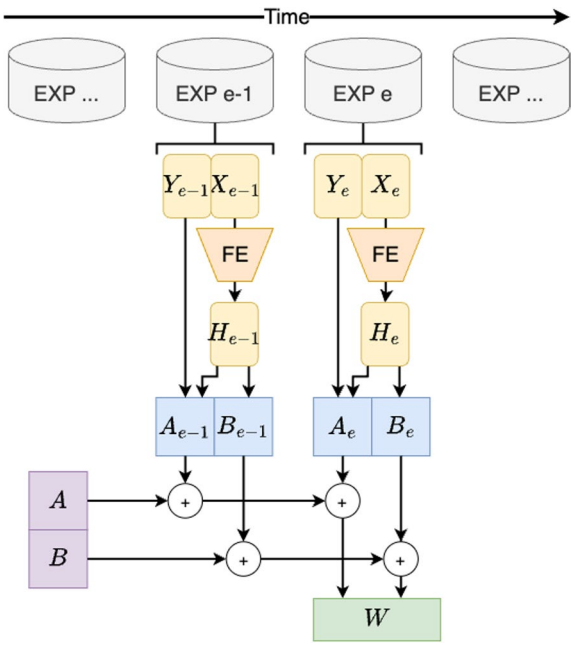


**Fig. 2** The PORR methodology entails fine-tuning a pre-trained feature extractor with RR, applied incrementally to maintain exact weights in the context of CL

## 4 Case study

This section describes materials and methods used to prepare the architecture pipeline and validate the proposed CL method, from selecting a feature extractor to the final accuracy analysis. Figure 3 represents a structured workflow outlining the key steps in developing the proposed CIL analysis pipeline, particularly when using a pre-trained model.

The process can be divided into essential steps that are useful to address the task objective. The first step involves defining the specific task and gathering the relevant datasets. The dataset is curated to cover various defect types and conditions the model encounters in training and CL scenarios. The second step concerns choosing suitable pre-trained models for feature extractors. These models embed the input data, providing the latent features necessary for subsequent processing. The third step analyzes the latest State-Of-The-Art techniques for CL methods, gathering and reviewing the different approaches. This helps identify the best methods to be applied for the selected specific task within the context of CL (Sect. 4.2). The fourth step is the method core, which regards the implementation of the PORR strategy. The fifth step focuses on developing the codebase necessary to support the analysis pipeline. The codebase will integrate the selected feature extraction, learning, and evaluation components to perform the continual learning task. In the sixth step, the appropriate model is chosen based on the requirements for solving the learning task, including accuracy, computational efficiency, and compatibility with CL methods. In the penultimate step of the process, the model undergoes a comprehensive assessment to evaluate its performance, ensuring that it meets the necessary benchmarks for CL and defect classification accuracy. Finally, the model's accuracy in detecting defects is analyzed across training epochs in the last step. This analysis provides insights into the model's learning progress and helps identify trends or areas for improvement during training.

### 4.1 Task and dataset

The two tasks addressed in this work are class-incremental learning, where one class is considered for each experience, and a simulation of an online defect detection task, involving smaller batches of data from one class followed by batches from another. The class-incremental problem is particularly challenging to solve because the model is trained on a new class each time, with entirely different input data statistics, which exacerbates catastrophic forgetting. On the

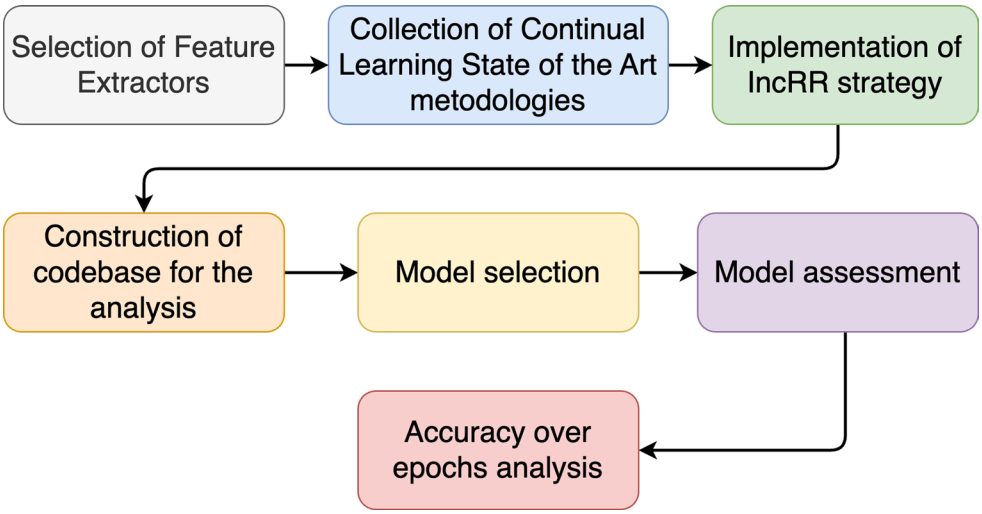


**Fig. 3** Operating Approach used to develop and demonstrate the proposed CIL method

other hand, an online variant of the class-incremental task is proposed for defect detection. Specifically, 30 experiences were structured, each containing 100 data points, to simulate a more online learning scenario.

#### 4.1.1 PBF of polymers dataset

The paper analyzes data from the case study reported in [14], concerning the manufacturing images of a PBF-LB/P powder bed, analyzed by Westphal and Seitz [15]. The images in this dataset were used to monitor and document the quality of the manufacturing process. Westphal and Seitz applied an ML approach to classify powder bed defects using VGG16 for feature extraction and the Xception CNN models with pretrained weights. They applied Transfer Learning, in which the features of an already trained CNN model are used to initialize the CNN used for the final classification. Figure 4 presents examples of defect (Fig. 4a) and free (Fig. 4b) images used for the training phase. The solid black area (manufactured element) in both images represents the component being manufactured, which corresponds to the component cross-section at that layer. The irregularities visible in Fig. 4a in the form of a horizontal line are caused by an incorrect distribution of polymer powders by the recoater.

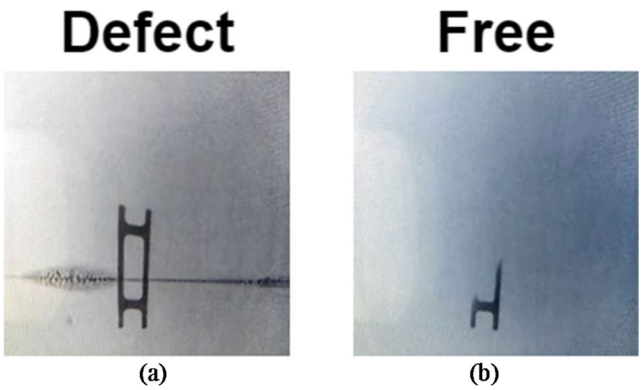


**Fig. 4** The categories "Defect" (**a**) and "Free" (**b**) indicate the presence or absence of defects. Westphal and Seitz published the analyzed dataset in [14] as data associated with [15]. The images in this figure are from [14] and have not been altered. The defects visible in (**a**) in the form of a horizontal line are caused by an incorrect distribution of polymer powders by the recoater

The original dataset is divided into two primary folders: the 'data' folder and the 'data_balanced' folder [14]. The 'data' folder contains 8,514 images, while the 'data_balanced' folder is more systematically structured, divided into three subgroups: training, validation_pbf, and test. Each subset is composed of two classes: "OK" (no defect) and "DEF" (defect). In particular, the training subgroup comprises 1,000 images per class, while the validation and test subgroups each consist of 500 images per class.

### 4.2 Benchmark methods

Five methods were selected from the state of the art to compare the performance of the proposed PORR approach. These methods were chosen due to their popularity, efficiency, and effectiveness, as they are widely recognized as robust approaches in most CL research. The continual learning library Avalanche [68] has been utilized to implement the code and utilize state-of-the-art benchmarks, which are listed and discussed below.

- Balanced Experience Replay and.
- GDumb.
- Learning without Forgetting.
- Elastic Weight Consolidation.
- Gradient Episodic Memory.

Experience Replay methods are characterized by their ability to store a small data buffer to train the model on both new

and old data, thereby avoiding catastrophic forgetting. The Balanced Experience Replay (BER) technique is a simple method for continually fine-tuning the model using new data and those in the replay memory [69]. Another technique is the Greedy Sampler and Dumb Learner (GDumb), a simple and effective method that is based on a greedy balancing sampler and a learner that each time is trained from scratch on the replay memory [70].

The Learning without Forgetting (LwF) method was developed to enable neural networks to acquire new competencies without impairing their performance on previously learned experiences, thus obviating the need to reuse the original training data [71]. LwF employs a function-regularization technique that allows learning from new experience while avoiding forgetting, using KD on the old model.

Elastic Weight Consolidation (EWC) is a weight regularization technique designed to mitigate catastrophic forgetting. It considers the importance of weights for a given experience [72]. The method employs a Fisher Information Matrix-based approach to identify and stabilize critical weights, enabling the network to learn new experiences without forgetting previous ones.

Gradient Episodic Memory (GEM) is an optimization-based technique that avoids catastrophic forgetting and allows the beneficial transfer of knowledge to previous experiences [73]. This is possible by comparing the new model gradients with the old model using the cosine similarity function. If the comparison results in gradients with similar directions, the latest model replaces the old one. In the case of conflicts, a model identical to the new one is searched for with gradients without conflicts.

### 4.3 Metrics

A comprehensive set of metrics was employed to evaluate the performance of the proposed approach (PORR) and other methodologies, ensuring a thorough assessment of classification accuracy and computational efficiency. Efficiency is measured by the method used to train the last layer, avoiding a pre-trained model that is fixed. The metrics employed for this evaluation are:

- The accuracy metric quantifies the proportion of test images that the model correctly classifies. This metric is crucial in determining the model's efficacy in accurately identifying classes or types of defects in datasets.
- Training time quantifies the total time (in seconds, s) required to apply a continuous methodology to a model on the training dataset. This metric is crucial for gauging a method's efficiency and suitability for real-time applications.
- The CPU Usage metric tracks the percentage of CPU (%) resources utilized when applying a CL methodology to a model in the training phase. This metric helps evaluate the computational burden imposed by the CL strategy. An Apple MacBook Pro equipped with an M1 10-core processor and 16GB of RAM has been used.
- Maximum RAM usage, in megabytes (MB), is a metric that monitors the maximum memory allocated during the continual learning process.
- Backward Transfer (BWT) evaluates the extent to which learning new experiences reinforces previously acquired knowledge. In this work, BWT is adapted to a single fixed test set evaluated after each incremental training epoch to accommodate the dataset and evaluation constraints. The metric is computed as:

$$BWT = \frac{1}{E-1}\sum_{e=1}^{E-1}(Acc_E - Acc_e) \tag{9}$$

where E is the total number of experiences, $Acc_e$ is the accuracy obtained after experience e, and $Acc_E$ is the final accuracy after the last experience. Higher values indicate stronger positive reinforcement and reduced forgetting.

Analyzing these metrics allows for a comprehensive understanding of the model's performance in terms of accuracy, speed, and resource efficiency.

### 4.4 Setting of hyperparameters

Model selection is a phase of the learning process in which the best hyperparameters are selected for the model and the learning methodology. To enhance the robustness of the analysis, the K-Fold cross-validation technique was employed, with K set to 4. In this way, we iteratively used the $i^{-th}$ positional fold as a validation set with the remaining folds as a training set, and, in the end, we took as the best hyperparameters those with the highest mean accuracy across the four folds. Table 4 shows the hyperparameters identified for the benchmark methods using the grid-search algorithm.

The number of training epochs was treated as a tunable hyperparameter and selected via validation for each strategy independently. Replay-based methods such as BER and GDumb were therefore evaluated over a wider range of epochs to analyze their convergence behavior, whereas the other strategies achieved stable performance with fewer epochs. This choice reflects the training paradigm of BER and GDumb, which repeatedly incorporate samples from previous experiences during optimization. The computational impact of these different configurations is explicitly reflected in the reported efficiency metrics (see Table 5).

**Table 4** Hyperparameters for each benchmark are considered in the validation phase

| Strategies | Hyperparameters |
|---|---|
| PORR | λ (regularization) = [0, 0.0001, 0.001, 1, 30, 50, 80, 100, 300, 500, 800, 1000] |
| BER | Learning rate = [0.01, 0.001, 0.1]<br>Weight decay (regularization) = 0.0001<br>Momentum = 0.9<br>Train epochs = [1, 5, 10]<br>Memory size = 50 |
| GDumb | Learning rate = [0.01, 0.001, 0.1]<br>Weight decay (regularization) = 0.0001<br>Momentum = 0.9<br>Train epochs = [1, 5, 10]<br>Memory size = 50 |
| LwF | Learning rate = [0.01, 0.001, 0.1]<br>Weight decay (regularization) = 0.0001<br>Momentum = 0.9<br>Train epochs = [1, 2, 3]<br>Alpha = [1, 0.1]<br>Temperature = [2, 1] |
| EWC | Learning rate = [0.01, 0.001, 0.1]<br>Weight decay (regularization) = 0.0001<br>Momentum = 0.9<br>Train epochs = [1, 2, 3]<br>Ewc lambda = [1, 0.1, 0.001] |
| GEM | Learning rate = [0.01, 0.001, 0.1]<br>Weight decay (regularization) = 0.0001<br>Momentum = 0.9<br>Train epochs = [1, 2, 3]<br>Patterns per experience = 50<br>Memory strength = 0.5 |

### 4.5 Model assessment

Model assessment is the last learning phase in which the model's performance is evaluated on the training and test sets. In this phase, the metric results discussed in Sect. 4.3 are computed to assess the performance of each method reported in Table 5. To ensure a more robust evaluation, the experiments were run 10 times, and the mean and standard deviation were collected for each metric (Fig. 5). The results are shown and discussed in the following section.

## 5 Results

This section reports the results for the analyzed dataset. Table 5 shows the metrics evaluated for the analyzed CIL methods using MNV3Large and MNV3Small as pre-trained FE models. Twelve combinations of techniques are explored, including the proposed PORR method. The six columns in this table refer to the base method, the pre-trained FEModel applied, and the calculated metrics: accuracy, training time, CPU usage during training, and the maximum RAM allocated during training. All rows are ordered from top to bottom in decreasing accuracy. The table reports the mean value and standard deviation for each metric. As already defined, these values are related to 10 trials to ensure greater robustness of the results.

**Table 5** Comparison between PORR and the other benchmark models for the analyzed PBF-LB/P dataset in terms of the four metrics described in Sect. 4.3, considering two pre-trained FEModels: MNV3Large and MNV3Small. Each metric is reported as the mean and standard deviation (in brackets) across 10 trials

| Method | FEModel | Accuracy | Training Time (s) | CPU Usage (%) | Max RAM Usage (MB) |
|---|---|---|---|---|---|
| GEM | MNV3Large | 95.62 (0.29) | 0.4183 (0.0872) | 761.35 (02.50) | 775.21 (9.84) |
| PORR | MNV3Large | 95.50 (0.0) | 0.0222 (0.0019) | 567.2 (49.13) | 704.94 (26.07) |
| BER | MNV3Large | 91.76 (2.12) | 1.7881 (0.0145) | 761.13 (4.35) | 750.31 (38.86) |
| GEM | MNV3Small | 91.6 (0.48) | 0.4062 (0.1296) | 763.90 (11.80) | 543.94 (80.81) |
| PORR | MNV3Small | 91.59 (00.01) | 0.0208 (0.0023) | 640.20 (40.83) | 612.79 (30.64) |
| BER | MNV3Small | 88.03 (01.47) | 1.8474 (0.0979) | 721.70 (49.96) | 518.21 (08.03) |
| GDumb | MNV3Large | 84.76 (03.16) | 1.7304 (0.0612) | 760.90 (15.02) | 790.40 (04.72) |
| GDumb | MNV3Small | 66.94 (11.07) | 1.7215 (0.0276) | 755.61 (29.70) | 488.22 (07.84) |
| EWC | MNV3Small | 60.61 (10.46) | 0.0133 (0.0013) | 676.95 (28.78) | 667.73 (02.76) |
| EWC | MNV3Large | 59.54 (08.28) | 0.0138 (0.0014) | 650.06 (50.93) | 740.75 (04.36) |
| LwF | MNV3Large | 50.0 (0.02) | 0.0211 (0.0014) | 582.25 (52.28) | 736.19 (05.44) |
| LwF | MNV3Small | 50.0 (0.05) | 0.0217 (0.0026) | 602.46 (46.11) | 656.99 (02.28) |

To enhance interpretability, Fig. 6 complements the numerical results by visually comparing accuracy, training time, CPU allocation, and memory requirements for each method. These plots allow a quick assessment of performance-efficiency trade-offs and highlight how different algorithms behave under identical computational constraints.

The results of the analyzed PBF-LB/P dataset show that GEM and PORR methods achieved higher performance in online class incremental tasks, achieving high accuracy values with the MNV3Large features extractor model. Even if GEM MNV3Large achieved the highest accuracy value, PORR with MNV3Large achieved a lower training time, the lowest CPU usage, and reduced RAM utilization. These two methods are followed in order: BER with MNV3Large, GEM with MNV3Small, and PORR with MNV3Small, which yield similar efficacy results. BER with MNV3Small and GDumb with MNV3Large have sufficient efficacy with

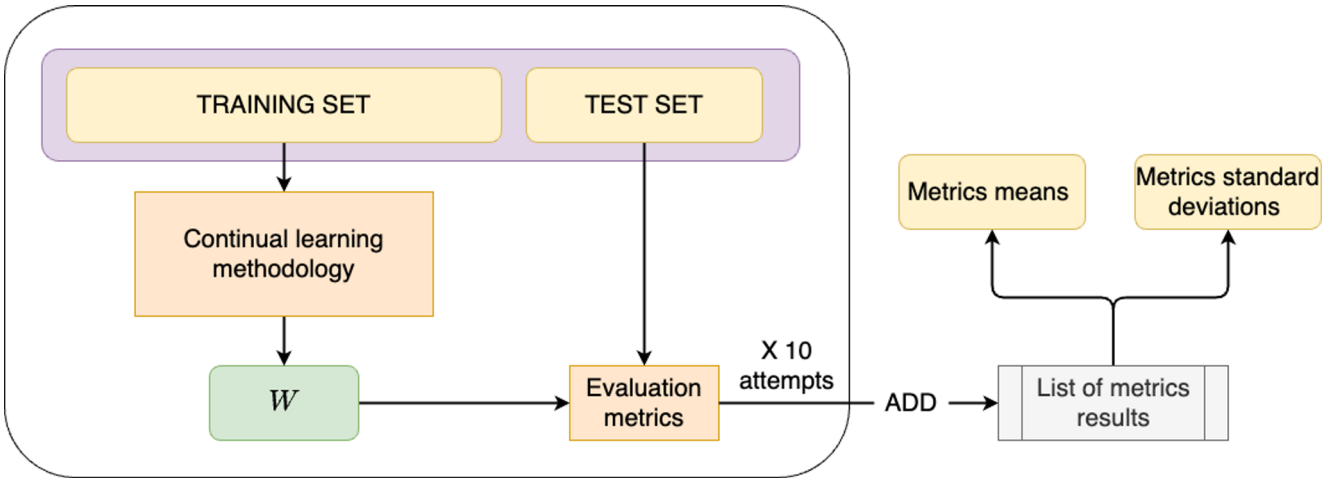


**Fig. 5** Model assessment: The model's performance in the described metrics is evaluated through repeated experiments (10 trials). The mean values and the standard deviation are collected for each metric

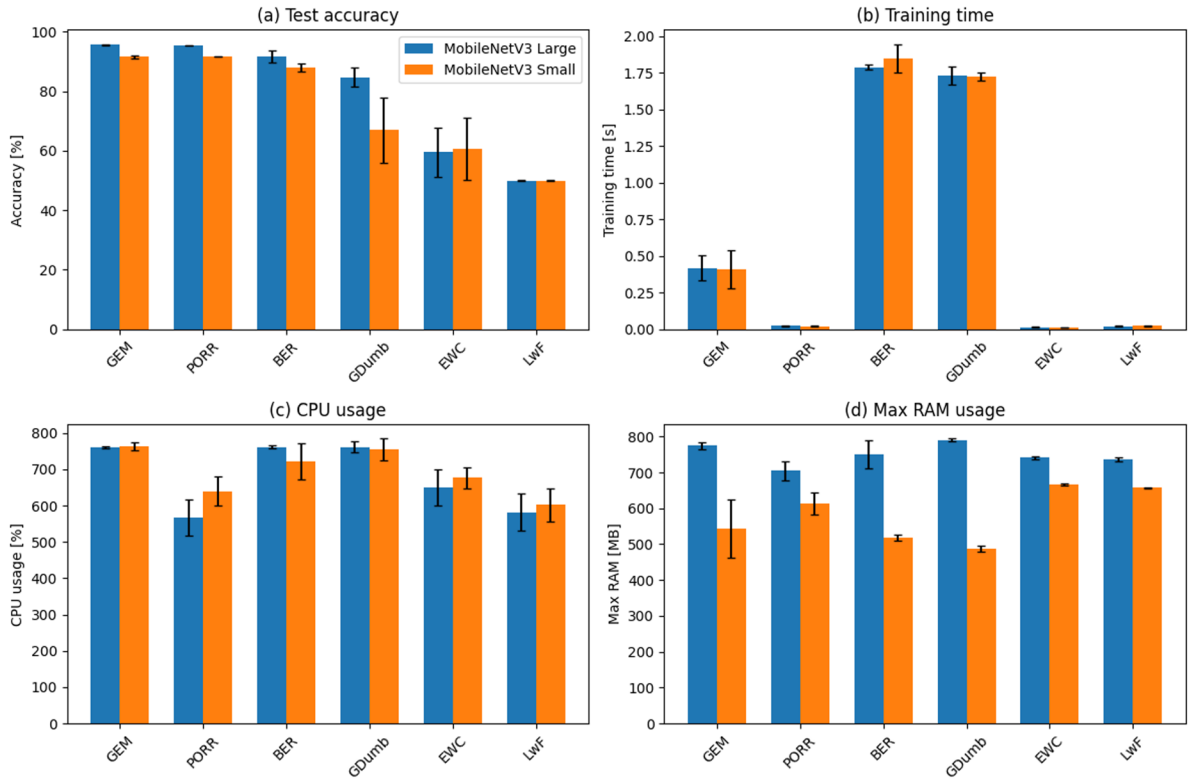


**Fig. 6** Comparative performance plots of the analyzed continual learning strategies, showing (**a**) mean accuracy, (**b**) mean training time, (**c**) mean CPU usage, and (**d**) maximum RAM consumption. Results are obtained using MobileNetV3 Small and MobileNetV3 Large as feature extractors over 10 experimental trials. These plots visually highlight the accuracy and computational trade-offs among PORR and state-of-the-art benchmark methods

a medium-level usage of computational resources. On the other hand, GDumb with MNV3Small, and all EWC and LwF methods, achieved lower performance, with an accuracy rate below 70% (LwF with MNV3Large and MNV3Small achieved an accuracy of 50%). This suggests these methodologies may not be well-suited for defect detection in the analyzed PBF-LB/P dataset.

Concluding, GEM and PORR with MNV3Large are the most promising strategies for the analyzed test case. Therefore, the results validate the proposed PORR method

for defect-recognition investigation by image analysis in PBF-LB/P.

### 5.1 Learning curve

The Accuracy after Each Experience (AEE) was evaluated considering 30 CL experiences, each including 100 data points. The resulting Learning Curve is reported in Fig. 7, where the plots are distinguished by MNV3Small (Fig. 7a) and MNV3Large (Fig. 7b) FEmodels.

PORR and GEM methods achieve the best accuracy with both models, with AEE between 90% and 95%. Moreover, the PORR method is more stable than GEM after the 18th experience, especially with the MNV3Small. The BER strategy underperforms slightly relative to PORR and GEM, exhibiting comparable stability. GDumb performs below these methods and displays less stable behavior, declining efficacy after the 18th experience due to catastrophic forgetting. Finally, LwF and EWC also demonstrate insufficient task performance. Overall, the increase in accuracy observed after the midpoint of the plot is expected: during the initial experiences, only data from a single class is available, whereas thereafter data from the other class is introduced.

### 5.2 Backward transfer analysis

To further evaluate the continual learning behavior of the analyzed methods, a Backward Transfer (BWT) analysis was performed. Unlike traditional settings, in which BWT is computed using per-task test sets, in this work, the metric was adapted to the experimental conditions by evaluating performance on a single held-out test set after each incremental training epoch. This approach maintains comparability across methods while providing insight into how newly acquired knowledge affects previously learned representations.

Figure 8 reports the resulting BWT values obtained using MobileNetV3-Large and MobileNetV3-Small as feature extractors. For improved interpretability, the numerical values are also summarized in Table 6. Higher BWT values indicate a beneficial transfer effect, meaning that learning new experiences results in improved or reinforced performance on previously learned knowledge. At the same time, lower scores reveal no transfer or negative interference.

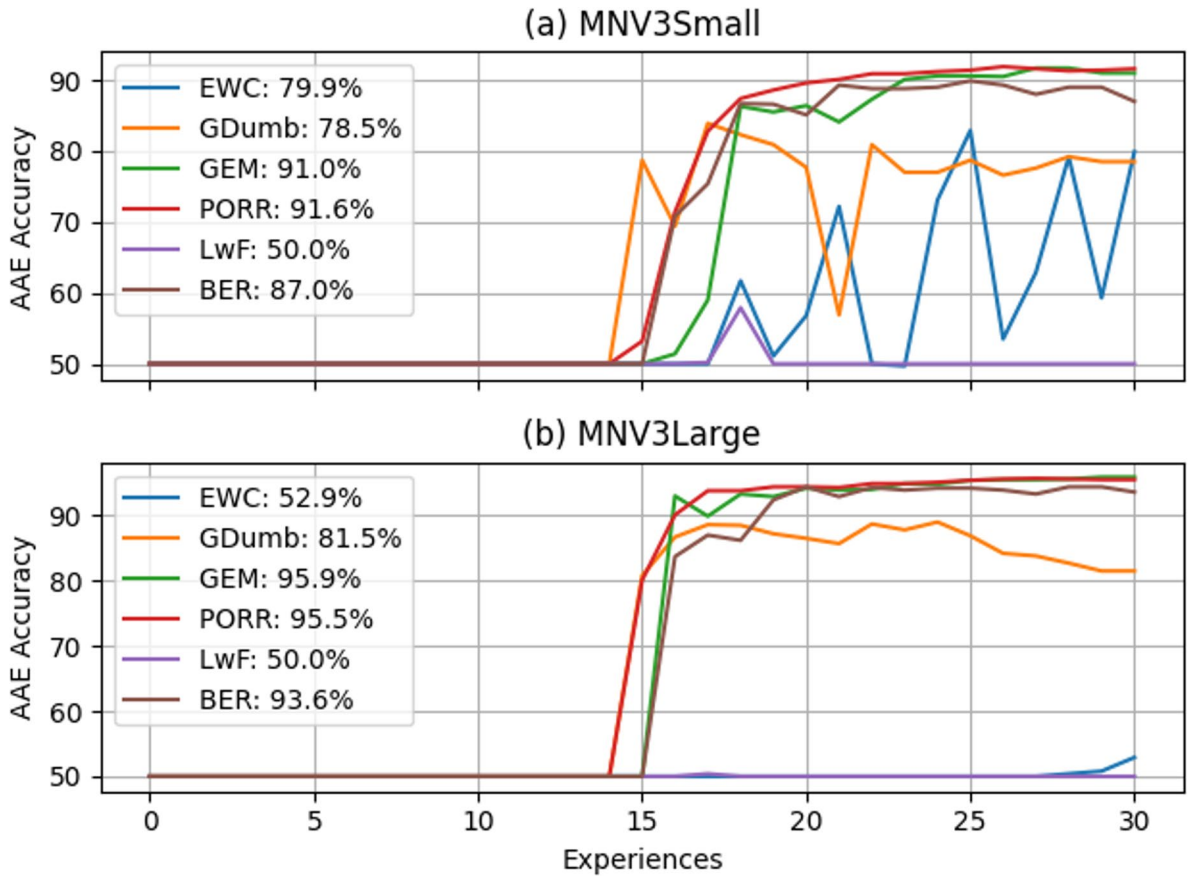


**Fig. 7** Learning Curve to describe the AEE behavior (y-axis) related to 30 CL experiences (x-axis) for MobileNetV3Small (Fig. 6a) and MobileNetV3Large (Fig. 6b) FEModels, considering all six methods analyzed in the proposed test case

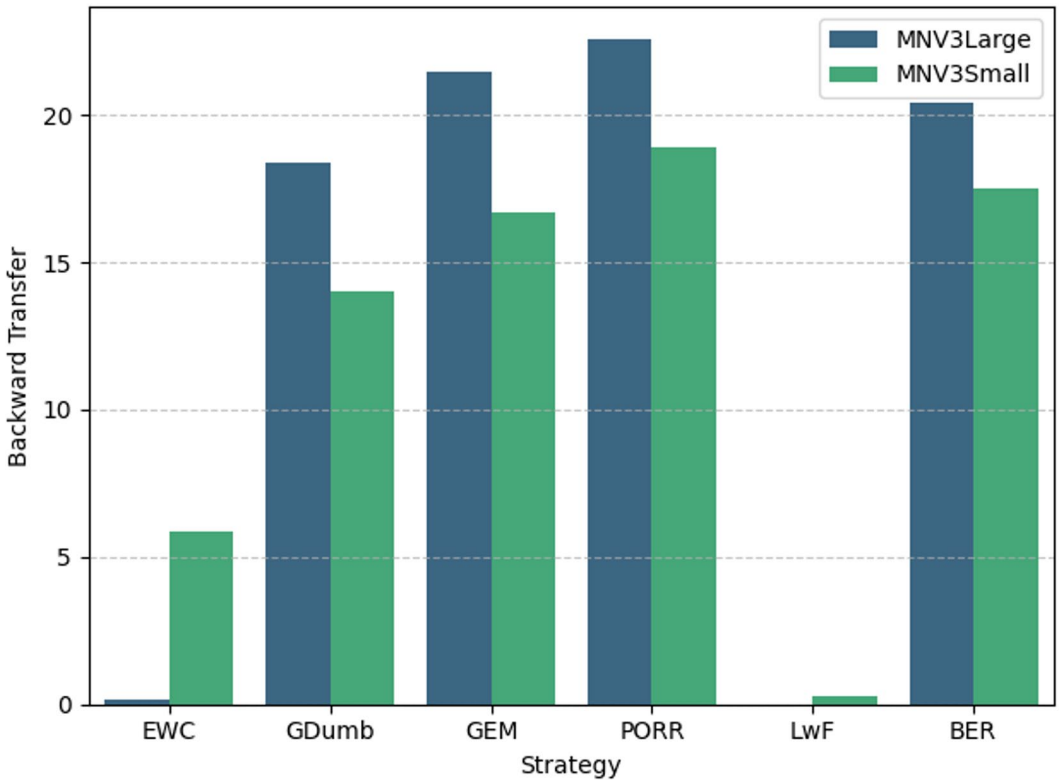


**Fig. 8** Backward Transfer comparison for the analyzed continual learning strategies using MobileNetV3-Large and MobileNetV3-Small feature extractors. BWT values are computed using a fixed test set evaluated after each incremental epoch to assess the positive or negative influence of newly learned experiences on previously acquired knowledge

**Table 6** Backward Transfer Comparison for Each Strategy and Feature Extractor

| Strategy | MobileNetV3-Large | MobileNetV3-Small |
|---|---|---|
| BER | 20.412903 | 17.506452 |
| EWC | 0.132258 | 5.880645 |
| GDumb | 18.390323 | 13.993548 |
| GEM | 21.441935 | 16.683871 |
| LwF | 0.012903 | 0.261290 |
| PORR | 22.551613 | 18.880645 |

Figure 8 shows that PORR achieves the highest BWT score across both feature extraction backbones, confirming that it not only preserves previously learned knowledge but also slightly reinforces it throughout incremental updates. GEM also demonstrates strong backward transfer performance, consistent with its gradient-based replay design. In contrast, LwF and EWC display minimal backward transfer, suggesting that their regularization strategies are insufficient under the examined class-incremental configuration. GDumb and BER show moderately high BWT values; however, BER exhibits reduced transfer when paired with MobileNetV3-Small, indicating dependency on feature representation richness.

These findings reinforce the conclusions drawn in Sect. 5: PORR demonstrates not only computational efficiency and accuracy stability, but also the most substantial positive knowledge reinforcement among all considered strategies.

## 6 Conclusion

The research leverages the employment of Machine Learning in Process Monitoring for AM. The main contributions of this work are the presentation of a Continual Learning methodology, called Progressive Online Ridge Regression (PORR), based on a fine-tuning technique for CNNs. The proposed method can learn episodically, achieving the same effectiveness as learning from the data in a single episode. The fine-tuning approach allows the number of iterations during learning to be reduced, starting from general knowledge of the data and applying Ridge Regression to fit the final layer placed on top of a pre-trained model. Fine-tuning the output layer for the given downstream task improves the ability to avoid catastrophic forgetting.

The analysis was conducted in the context of real-time defect detection, a crucial issue in AM during the build

stage. A dataset containing defects related to the PBF-LB/P processes [14] was used to validate a computational tool to improve online defect detection and control of the manufacturing process. Early defect detection can reduce process costs and time while improving process quality. The results confirm the feasibility of implementing a Continual Learning approach in AM. Furthermore, the obtained results could be implemented in tools used in the embodiment design phase to improve the AM production, reducing the possibility of defect-related failures.

Analyzing the performance of the proposed ML approach, the work demonstrates the importance of selecting an appropriate CL strategy and a Feature Extraction model tailored to the dataset's characteristics and the constraints of the computational environment in which the application runs. The tasks addressed include class incremental learning with distinct experiences for each class and an online defect detection task with smaller batches, underscoring the complexity of managing catastrophic forgetting. An image classification dataset was employed for defect recognition in PBF-LB/P, and a comprehensive set of metrics was used to evaluate accuracy, training time, CPU usage, and maximum RAM usage. The proposed PORR strategy exhibits optimal performance across the datasets, demonstrating superior accuracy, minimal training time, and efficient resource utilization. About Feature Extraction models, the MobileNetV3 Small model demonstrates effectiveness in resource-limited environments, whereas the MobileNetV3 Large model exhibits superior accuracy with greater computational resources. Focusing on benchmarks, GEM and GDumb strategies also demonstrate satisfactory performance compared to PORR, despite higher computational costs.

In contrast, the BER, LwF, and EWC strategies demonstrate suboptimal performance relative to PORR, indicating the need for further optimization for these specific datasets. The comprehensive model assessment phase, which includes repeated experiments, ensures robust results. This emphasizes the importance of strategic model selection and CL methodologies for achieving high performance across various defect-detection contexts. As future work, the resulting model will be tested in other defect-recognition applications for AM to enhance ML for Process Monitoring.

**Authors' contributions** Michele Trovato: Conceptualization, Writing; Mariorosario Prist: Conceptualization, Writing, Methodology, Investigation, Supervision; Paolo Cicconi: Conceptualization, Methodology, Writing, Supervision; Andrea Bonci: Supervision; Geremia Pompei: Writing, Development, Methodology; Lorenzo Longarini: Writing, Development, Methodology.

**Funding** Open access funding provided by Università degli Studi Roma Tre within the CRUI-CARE Agreement. No funding.

**Data availability** Data will be made available on request.

## Declarations

**Consent for publication** All authors have read and agreed to the published version of the manuscript.

**Conflict of interest** The authors declare no conflicts of interest.



## References

1. Huang SH, Liu P, Mokasdar A, Hou L (2012) Additive manufacturing and its societal impact: a literature review. Int J Adv Manuf Technol 67(5–8):1191–1203. https://doi.org/10.1007/s00170-012-4558-5
2. Jin Z, Zhang Z, Demir K, Gu GX (2020) Machine learning for advanced additive manufacturing. Matter 3(5):1541–1556. https://doi.org/10.1016/j.matt.2020.08.023M
3. Wang C, Tan XP, Tor SB, Lim CS (2020) Machine learning in additive manufacturing: state-of-the-art and perspectives. Addit Manuf 36:101538. https://doi.org/10.1016/j.addma.2020.101538
4. Caggiano A, Zhang J, Alfieri V, Caiazzo F, Gao R, Teti R (2019) Machine learning-based image processing for on-line defect recognition in additive manufacturing. CIRP Ann 68(1):451–454. https://doi.org/10.1016/j.cirp.2019.03.021
5. Zhang Y, Hong GS, Ye D, Zhu K, Fuh JYH (2018) Extraction and evaluation of melt pool, plume and spatter information for powder-bed fusion AM process monitoring. Mater amp; Des 156:458–469. https://doi.org/10.1016/j.matdes.2018.07.002
6. Rigolli D, Pompei L, Manfredini M, Vignoli M, La Battaglia V, Giorgetti A (2025) The development of a predictive maintenance system for gearboxes through a statistical diagnostic analysis of lubricating oil and artificial intelligence. Machines 13(8):693. https://doi.org/10.3390/machines13080693
7. Shin H, Lee JK, Kim J, Kim J (2017) Continual learning with deep generative replay. Advances in neural information processing systems, 30
8. Tercan H, Deibert P, Meisen T (2022) Continual learning of neural networks for quality prediction in production using memory aware synapses and weight transfer. Journal of Intelligent Manufacturing, vol. 33, no. 1, pp. 283–292, 2022
9. McDonnell MDT, Arnaldo D, Pelletier E, Grant-Jacob JA, Praeger M, Karnakis D, Eason RW, Mills B (2021) Machine learning for multi-dimensional optimisation and predictive visualisation of laser machining. J Intell Manuf 32(5):1471–1483. https://doi.org/10.1007/s10845-020-01717-4
10. Ma D, Jiang P, Shu L, Gong Z, Wang Y, Geng S (2022) Online porosity prediction in laser welding of aluminum alloys based on a multi-fidelity deep learning framework. J Intell Manuf 35(1):55–73. https://doi.org/10.1007/s10845-022-02033-9

11. Sun W, Al Kontar R, Jin J, Jionghua, Chang T-S (2023) A continual learning framework for adaptive defect classification and inspection. J Qual Technol 55(5):598–614. https://doi.org/10.1080/00224065.2023.2224974
12. Hoerl AE, Kennard RW (1970) Ridge regression: biased estimation for nonorthogonal problems. Technometrics 12(1):55–67. https://doi.org/10.2307/1267351
13. Divyam Madaan J, Yoon Y, Li (2021) Yunxin Liu, and Sung Ju Hwang. Representational continuity for unsupervised continual learning. In International Conference on Learning Representations
14. Westphal E, Seitz H (2021) SLS Powder bed defects. Mendeley Data V1. https://doi.org/10.17632/2yzjmp52fw.1
15. Westphal E, Seitz H (2021) A machine learning method for defect detection and visualization in selective laser sintering based on convolutional neural networks. Addit Manuf 41:101965. https://doi.org/10.1016/j.addma.2021.101965
16. Dilberoglu UM, Gharehpapagh B, Yaman U, Dolen M (2017) The role of additive manufacturing in the era of Industry 4.0. Procedia Manuf 11:545–554. https://doi.org/10.1016/j.promfg.2017.07.148
17. Mies D, Marsden W, Warde S (2016) Overview of additive manufacturing informatics: a digital thread. Integr Mater Manuf Innov 5(1):114–142. https://doi.org/10.1186/s40192-016-0050-7
18. Zhou L, Miller J, Vezza J, Mayster M, Raffay M, Justice Q, Al Tamimi Z, Hansotte G, Sunkara LD, Bernat J (2024) Additive manufacturing: a comprehensive review. Sensors 24(9):2668. https://doi.org/10.3390/s24092668
19. Sbrugnera Sotomayor NA, Caiazzo F, Alfieri V (2021) Enhancing design for additive manufacturing workflow: optimization, design and simulation tools. Appl Sci 11(14):6628. https://doi.org/10.3390/app11146628
20. Krueckemeier S, Anderl R, Schleich B, FOR EFFICIENT DIGITAL PROCESS CHAINS IN ADDITIVE MANUFACTURING (2023) FILE FORMAT SELECTION. Proceedings of the Design Society, 3, 1875–1884. https://doi.org/10.1017/pds.2023.188
21. Standard UNI EN ISO/ASTM 52900:2022
22. Trovato M, Belluomo L, Bici M, Campana F, Cicconi P (2024) Machine Learning Trends in Design for Additive Manufacturing. Des Tools Methods Industrial Eng III 109–117. https://doi.org/10.1007/978-3-031-52075-4_14
23. Trovato M, Cicconi P (2023) Design tools for metal additive manufacturing: a critical and perspective overview. Procedia CIRP 119:1084–1090. https://doi.org/10.1016/j.procir.2023.03.151
24. Kafle A, Luis E, Silwal R, Pan HM, Shrestha PL, Bastola AK (2021) 3D/4D printing of polymers: fused deposition modelling (FDM), selective laser sintering (SLS), and stereolithography (SLA). Polymers 13(18):3101. https://doi.org/10.3390/polym13183101
25. Papazoglou EL, Karkalos NE, Karmiris-Obratański P, Markopoulos AP (2021) On the modeling and simulation of SLM and SLS for metal and polymer powders: a review. Arch Comput Methods Eng 29(2):941–973. https://doi.org/10.1007/s11831-021-09601-x
26. Hofland EC, Baran I, Wismeijer DA (2017) Correlation of process parameters with mechanical properties of laser sintered PA12 parts. Adv Mater Sci Eng 2017:1–11. https://doi.org/10.1155/2017/4953173
27. Rahman MM, Ahmed KA, Karim M, Hassan J, Roy R, Bustami B, Alam SMN, Younes H (2023) Optimization of selective laser sintering three-dimensional printing of thermoplastic polyurethane elastomer: a statistical approach. J Manuf Mater Process 7(4):144. https://doi.org/10.3390/jmmp7040144
28. Han W, Kong L, Xu M (2022) Advances in selective laser sintering of polymers. Int J Extrem Manuf 4(4):042002. https://doi.org/10.1088/2631-7990/ac9096
29. Idriss AI, Li J, Guo Y, Wang Y, Li X, Zhang Z, Elfaki EA (2020) Sintering quality and parameters optimization of sisal fiber/PES composite fabricated by selective laser sintering (SLS). J Thermoplast Compos Mater 35(10):1632–1646. https://doi.org/10.1177/0892705720939179
30. Zhang H, LeBlanc S (2018) Processing parameters for selective laser sintering or melting of oxide ceramics. Additive Manuf High-Performance Met Alloys - Model Optim. https://doi.org/10.5772/intechopen.75832
31. Yehia HM, Hamada A, Sebaey TA, Abd-Elaziem W (2024) Selective laser sintering of polymers: process parameters, machine learning approaches, and future directions. J Manuf Mater Process 8(5):197. https://doi.org/10.3390/jmmp8050197
32. Trovato M, Belluomo L, Bici M, Prist M, Campana F, Cicconi P (2025) Machine learning in design for additive manufacturing: a state-of-the-art discussion for a support tool in product design lifecycle. Int J Adv Manuf Technol 137(5–6):2157–2180. https://doi.org/10.1007/s00170-025-15273-9
33. Pandiyan V, Masinelli G, Claire N, Le-Quang T, Hamidi-Nasab M, de Formanoir C, Esmaeilzadeh R, Goel S, Marone F, Logé R, Van Petegem S, Wasmer K (2022) Deep learning-based monitoring of laser powder bed fusion process on variable time-scales using heterogeneous sensing and operando X-ray radiography guidance. Addit Manuf 58:103007. https://doi.org/10.1016/j.addma.2022.103007
34. Ye D, Hong GS, Zhang Y, Zhu K, Fuh JYH (2018) Defect detection in selective laser melting technology by acoustic signals with deep belief networks. Int J Adv Manuf Technol 96(5–8):2791–2801. https://doi.org/10.1007/s00170-018-1728-0
35. Klamert V, Schmid-Kietreiber M, Bublin M (2022) A deep learning approach for real time process monitoring and curling defect detection in selective laser sintering by infrared thermography and convolutional neural networks. Procedia CIRP 111:317–320. https://doi.org/10.1016/j.procir.2022.08.030
36. Luo S, Ma X, Xu J, Li M, Cao L (2021) Deep learning based monitoring of spatter behavior by the acoustic signal in selective laser melting. Sensors 21(21):7179. https://doi.org/10.3390/s21217179
37. Scime L, Beuth J (2018) Anomaly detection and classification in a laser powder bed additive manufacturing process using a trained computer vision algorithm. Addit Manuf 19:114–126. https://doi.org/10.1016/j.addma.2017.11.009
38. Sahar T, Rauf M, Murtaza A, Khan LA, Ayub H, Jameel SM, Ahad IU (2023) Anomaly detection in laser powder bed fusion using machine learning: a review. Results Eng 17:100803. https://doi.org/10.1016/j.rineng.2022.100803
39. Grasso M, Remani A, Dickins A, Colosimo BM, Leach RK (2021) In-situ measurement and monitoring methods for metal powder bed fusion: an updated review. Meas Sci Technol 32(11):112001. https://doi.org/10.1088/1361-6501/ac0b6b
40. EOS GmbH Electro Optical Systems. (n.d.). EOS Smart Monitoring – Smart Fusion. Retrieved February 20 (2026) from https://www.eos.info/enablement/software/eos-smart-monitoring/smart-fusion
41. Additive Assurance (n.d.). AMiRIS®: Best in class quality assurance for L-PBF. Retrieved February 20, 2026, from https://www.additiveassurance.com/product.html
42. OctoPrint Community. (n.d.). OctoPrint. Retrieved February 25 (2026) from https://octoprint.org/
43. OctoPrint Plugin Repository. (n.d.). Obico for OctoPrint. Retrieved February 20 (2026) from https://plugins.octoprint.org/plugins/obico/
44. Markforged. (n.d.). Introducing Blacksmith: The Adaptive Manufacturing Platform. Retrieved February 20 (2026) from https://markforged.com/resources/blog/introducing-blacksmith
45. 3D Systems. (n.d.). SLS 380 | Industrial selective laser sintering system. Retrieved February 22 (2026) from https://www.3dsystems.com/3d-printers/sls-380

46. Formlabs. (n.d.). Fuse 1 – Stampante 3D SLS compatta. Retrieved February 22 (2026) from https://formlabs.com/it/3d-printers/fuse-1/
47. Inkbit. (n.d.). Technology. Retrieved February 22 (2026) from https://www.inkbit3d.com/technology
48. EOS GmbH Electro Optical Systems (2026) FORMIGA P 110 Velocis – Scheda tecnica. Retrieved February 22, 2026, from https://www.eos.info/it/soluzioni-polimeriche/stampanti-polimeri/schede-tecniche/sds-formiga-p-110-velocis
49. Kudithipudi D, Aguilar-Simon M, Babb J, Bazhenov M, Blackiston D, Bongard J, Brna AP, Chakravarthi Raja S, Cheney N, Clune J, Daram A, Fusi S, Helfer P, Kay L, Ketz N, Kira Z, Kolouri S, Krichmar JL, Kriegman S, Levin M, Madireddy S, Manicka S, Marjaninejad A, McNaughton B, Miikkulainen R, Navratilova Z, Pandit T, Parker A, Pilly PK, Risi S, Sejnowski TJ, Soltoggio A, Soures N, Tolias AS, Urbina-Meléndez D, Valero-Cuevas FJ, van de Ven GM, Vogelstein JT, Wang F, Weiss R, Yanguas-Gil A, Zou X, Siegelmann H (2022) Biological underpinnings for lifelong learning machines. Nat Mach Intell 4(3):196–210. https://doi.org/10.1038/s42256-022-00452-0
50. Wang L, Zhang X, Su H, Zhu J (2024) A comprehensive survey of continual learning: theory, method and application. IEEE Trans Pattern Anal Mach Intell 46(8):5362–5383. https://doi.org/10.1109/tpami.2024.3367329
51. van de Ven GM, Tuytelaars T, Tolias AS (2022) Three types of incremental learning. Nat Mach Intell 4(12):1185–1197. https://doi.org/10.1038/s42256-022-00568-3
52. Peng D Learning to generalize to new tasks/domains with limited data. https://doi.org/10.32657/10356/171769
53. Li Y, Xie T, Liu C, Shi Z (2024) Pseudo replay-based class continual learning for online new category anomaly detection in advanced manufacturing. IISE Trans. https://doi.org/10.1080/24725854.2024.2428642
54. Arslan Chaudhry M, Rohrbach M, Elhoseiny T, Ajanthan PK, Dokania PHS (2019) Torr, and Marc'Aurelio Ranzato. On tiny episodic memories in continual learning. arXiv preprint arXiv:1902.10486
55. Hanul Shin JK, Lee J, Kim, Kim J (2017) Continual learning with deep generative replay. Adv Neural Inf Process Syst, 30
56. Xialei Liu C, Wu M, Menta L, Herranz B, Raducanu AD, Bagdanov (2020) Shangling Jui, and Joost van de Weijer. Generative feature replay for class-incremental learning. CVPR Workshops, pages 226–227
57. Kirkpatrick J, Pascanu R, Rabinowitz N, Veness J, Desjardins G, Rusu AA, Milan K, Quan J et al (2017) Tiago Ramalho, Agnieszka Grabska-Barwinska,. Overcoming catastrophic forgetting in neural networks. Proceedings of the National Academy of Sciences, 114(13):3521–3526
58. Hippolyt Ritter A, Botev, Barber D (2018) Online structured laplace approximations for overcoming catastrophic forgetting. Adv Neural Inf Process Syst 31
59. Gou J, Yu B, Maybank SJ, Tao D (2021) Knowledge distillation: a survey. Int J Comput Vis 129(6):1789–1819
60. Arslan Chaudhry Marc'Aurelio, Ranzato MR, Elhoseiny M (2018) Efficient lifelong learning with a-gem. In International Conference on Learning Representations
61. Mehta SV, Patil D, Chandar S, Strubell E (2021) An empirical investigation of the role of pre-training in lifelong learning. arXiv preprint arXiv:2112.09153
62. Mallya A, Davis D, Lazebnik S (2018) Piggyback: Adapting a single network to multiple tasks by learning to mask weights. In Proceedings of the European Conference on Computer Vision, pages 67–82
63. Ziyang Wu C, Baek CY, Ma Y (2021) Incremental learning via rate reduction. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 1125–1133
64. Andrei A, Rusu NC, Rabinowitz G, Desjardins H, Soyer James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. Progressive neural networks. arXiv preprint arXiv:1606.04671, 2016.
65. Andrew G, Howard, Zhu M, Chen B (2017) and Dmitry Kalenichenko and Weijun Wang and Tobias Weyand and Marco Andreetto and Hartwig Adam, MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications, journal of CoRR}, vol. abs/1704.04861
66. Howard A, Sandler M, Chen B, Wang W, Chen L, Tan M, Chu G, Vasudevan V, Zhu Y, Pang R, Adam H, Le Q (2019) Searching for MobileNetV3, 2019 IEEE/CVF International Conference on Computer Vision (ICCV), pp. 1314–1324
67. De Caro V, Gallicchio C, Bacciu D (2023) Continual adaptation of federated reservoirs in pervasive environments. Neurocomputing 556:126638
68. Lomonaco V et al (2021) Avalanche: a n End-to-End Library for Continual Learning, 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Nashville, TN, USA, pp. 3595–3605. https://doi.org/10.1109/CVPRW53098.2021.00399
69. Sun H, Li R, Yang H, Zhu W Balanced Prioritized Experience Replay, 2022 3rd International Conference on Electronic Communication and Artificial Intelligence (IWECAI), Zhuhai, China, 2022, pp. 200–203. https://doi.org/10.1109/IWECAI55315.2022.00046
70. Prabhu A, Torr PHS, Dokania PK (2020) GDumb: A Simple Approach that Questions Our Progress in Continual Learning. In: Vedaldi A, Bischof H, Brox T, Frahm JM (eds) Computer Vision – ECCV 2020. ECCV 2020. Lecture Notes in Computer Science, vol 12347. Springer, Cham. https://doi.org/10.1007/978-3-030-58536-5_31
71. Li Z, Hoiem D (2018) Learning without Forgetting. IEEE Trans Pattern Anal Mach Intell 40(12):2935–2947. https://doi.org/10.1109/TPAMI.2017.2773081
72. Kirkpatrick J et al (2017) Overcoming catastrophic forgetting in neural networks. Proc Natl Acad Sci USA 114:3521–3526
73. Lopez-Paz (2017) David, and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. Adv Neural Inf Process Syst 30