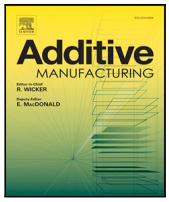
Additive Manufacturing 89 (2024) 104271

Contents lists available at ScienceDirect

Additive Manufacturing

journal homepage: www.elsevier.com/locate/addma



Research paper

Iterative learning for efficient additive mass production

Christos Margadji, Douglas A.J. Brion[1], Sebastian W. Pattinson *

Department of Engineering, University of Cambridge, Trumpington Street, Cambridge, United Kingdom

ARTICLE INFO

Keywords:
Mass production
Process control
Artificial intelligence
Deep learning

ABSTRACT

Material extrusion could enable on-demand production of complex and personalized parts but is limited by low reliability, particularly in higher-volume production. Machine learning-based methods may enhance reliability, but are often themselves insufficiently reliable for use in production. Foundation artificial intelligence models have enabled significant improvements in performance across many tasks. Here, a vision-based control system is reported, coupling active learning and uncertainty awareness with a foundation model to continually learn to build a specific part better. The resulting framework is called Iterative Learning, as it improves performance by learning from its own errors during repeated build cycles of the same part. The iterative learning approach is shown to enable robust error detection and correction while being more space, time and computationally efficient compared to a naive fine-tuning approach. This provides a path showing how foundation models may be adapted to enhance reliability across a wider range of additive manufacturing processes.

1. Introduction

Material extrusion, where material is selectively dispensed through a nozzle at predetermined locations within the build volume, is the most widespread additive manufacturing (AM) method for reasons including its low-cost, ease of use and compatibility with diverse polymers, metals, ceramics and other materials. Typically, feedstock in the form of filament is heated and then extruded through a nozzle onto the build surface in a layer-by-layer fashion [1]. This unique approach can produce complex and customized products where and when they are needed, potentially leading to novel products and industrial systems across diverse applications such as aerospace, medical devices and construction [2–4]. However, material extrusion and other AM systems struggle in higher volume production. A significant cause of this difficulty is AM's propensity for errors caused by factors including variability in feedstock, fluctuations in build chamber conditions, differences between AM machines, and the physical complexity of the AM process itself [5]. These challenges can frequently lead to unrecoverable build cycles that waste material, energy, and time.

Similarly to many other manufacturing processes, the current error mitigation strategy in AM relies on expert human operators who manually adjust the process parameters using a trial and error approach [6]. However, manual intervention becomes problematic when required for prolonged hours, when many machines are being used simultaneously, or when real-time interventions are needed. These challenges are poised to grow with the advancement of AM, particularly as more difficult materials are being handled in more challenging environments [7–9].

As an alternative to manual monitoring, there has been a surge of interest in developing intelligent error detection and correction systems. Specifically for material extrusion, acoustic, inertial, pressure and current sensors have been used for process monitoring, enabling detection of anomalies such as nozzle clogs [10–16]. But data from such sensors tend not to be rich enough to enable comprehensive error detection and correction. Vision sensors are more information-rich, enabling the detection of larger scale defects such as layer shifts and low-quality infills even with traditional computer vision methods [17–19]. Multi-camera systems, offering insights not visible from a single visible-spectrum camera, have also been explored [17,20,21]. Recently, deep learning has emerged as a promising route for monitoring AM processes and overcoming the limitations of handcrafted features for error detection [22,23]. Coupled with control, errors could also be mitigated by optimizing the process parameters post-build or controlling them in-situ [24–27]. However, deep learning techniques remain challenging to implement in practice, for reasons including instabilities in accuracy as well as the large amount of task-specific training data required. Importantly, instabilities in accuracy introduce noise in the controlled parameter, resulting in sub-optimal corrections. The recent rise of foundation AI models has led to significantly improved performance in many domains [28]. A foundation model is trained on a diverse range of tasks using large datasets, enabling it to

* Corresponding author.
E-mail address: swp29@cam.ac.uk (S.W. Pattinson).
[1] Present address: Matta Labs, East Road, London, United Kingdom.

https://doi.org/10.1016/j.addma.2024.104271
Received 8 April 2024; Received in revised form 28 May 2024; Accepted 18 June 2024
Available online 21 June 2024

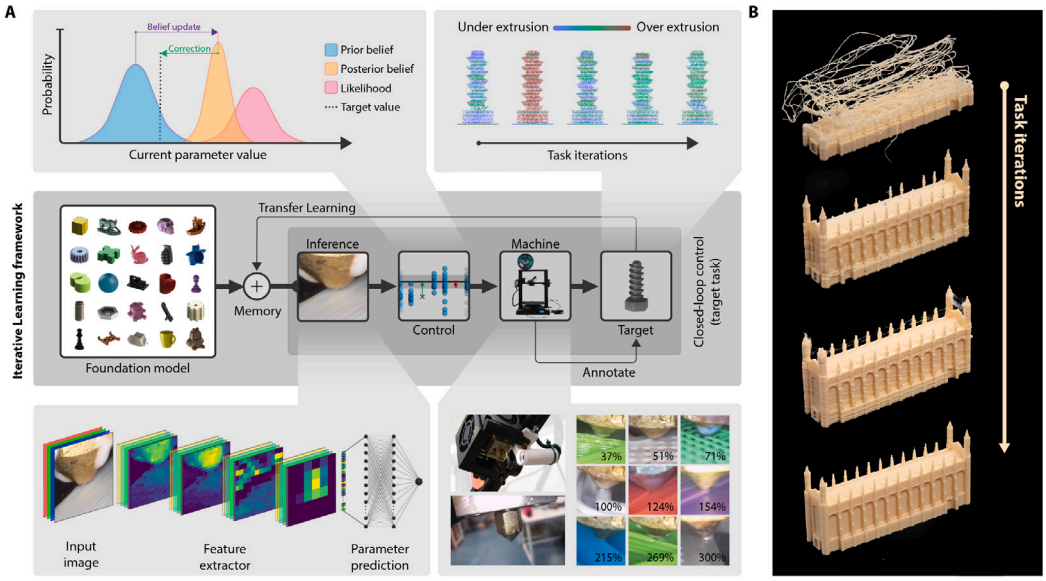


Fig. 1. Overview of the iterative learning framework. A. Schematic of the framework and graphical summaries of the sub-tasks involved. The foundation model is initially used to control the target task. New data are acquired, labeled via the machine-in-the-loop and stored in memory. Before subsequent iterations, the union of existing knowledge from all prior episodes is used to retrain the agent, making it increasingly capable in performing the task in-hand. B. Demonstration of the framework showing progressive enhancement in part precision as more iterations of the target task are performed.

generalize and perform well in new situations with minimal additional training. However, the applicability of foundation models and how these can be adapted best in manufacturing scenarios remains relatively under-explored.

In this work, a vision-based control system is presented which integrates active learning with a foundation model. This model leverages its own predictive uncertainty to progressively specialize in the mass production of a specific part, as illustrated in Fig. 1. Mass production refers to large-scale manufacturing processes where a high volume of identical products is produced using automated systems. The developed approach, called iterative learning (IL) hereafter, uniquely advances the system's capability, allowing it to learn from every build cycle as well as from the varying environmental conditions it encounters. Using IL, the system is able to make use of the continual data inflow from mass production, and therefore to demonstrate increased efficiency and precision in recovering from errors if and when they occur.

2. Methods

2.1. Data collection and pre-processing

The experimental setup shown in Figure S1 was configured for data collection. It consisted of a Creality CR-20 Pro material extrusion AM system, equipped with a commercial endoscope (Pancellent 2.0 Megapixel CMOS camera) attached on its extruder head. No additional modifications were made to the AM system, to replicate typical operational conditions. The endoscope was directed towards the extrusion nozzle, facilitating real-time monitoring of the material deposition within a moving frame of reference. Both the AM system and the endoscope were interfaced with a Raspberry Pi 4 Model B running OctoPrint 1.9.3, utilizing a custom plugin adapted from [24]. This plugin facilitated communication with an online server by posting requests towards its IP address. The server was responsible for receiving these requests and storing the corresponding content locally. Every request included an image captured by the mounted endoscope, alongside labels obtained from the printer's firmware, including the material flow rate at the instance of collection. The material flow rate, defined as the percentage of material exiting the nozzle's orifice per unit time, was directly acquired from the printer's firmware. This measurement is always relative to the default settings established during calibration. The AM system utilized polylactic acid (PLA) material feedstock in the form of 1.75 mm filament, sourced from various manufacturers such as PolyMaker, colorFabb, and Fillamentum, unless specified otherwise.

To introduce a diverse range of potential defects in the dataset, deliberate flow rate errors were induced during each build cycle. After sourcing parts from Thingiverse or creating them using Autodesk Fusion 360, and subsequently slicing them using Ultimaker 5.2.1, a Python script was employed to designate certain layers as defective. Defective layers were initiated by adjusting the flow rate value between 30% and 300% using the `M221` G-code command. To ensure a balanced dataset, adjustments within this range were selected from a predefined list of 30 evenly spaced values. Once a value was selected, it was removed from the list, preventing its reselection until all other remaining values had been used. To mitigate biases introduced by prior defects, healthy layers were deposited on top of defective ones to serve as primers. While this was typically done on a layer-by-layer basis, parts featuring infill types with alternating hatch orientations required two healthy layers to ensure that the data with defects included both orientations. Samples collected during the deposition of primer layers were excluded from the dataset to avoid bias towards good samples.

At the default sampling rate of 15 Hz, high similarity between consecutive frames was observed, posing two significant challenges. Firstly, it increased the risk of model over fitting, as redundant data may impede the learning process of the AI system. Secondly, it compromised the integrity of the train/validation/test split, as identical images across these subsets may violate the principle of evaluating models on unseen data, thus undermining the accuracy of the reported model assessments. To address these issues, all collected data were down sampled to a frame rate equivalent of 3 Hz. This adjustment allowed a larger time gap for visual changes between frames to be observable. The choice of the down-sampling factor was based on a qualitative assessment of various consecutive frame pairs.

Additionally, images captured immediately following a flow rate adjustment were disregarded until the system stabilized at the specified value for the defective layer. This was done to eliminate any samples that may have been erroneously labeled due to software or mechanical delays in applying the process parameter changes. To quantitatively assess the time required for changes to become visually apparent, 20 tests were conducted where the flow rate was varied between 30% to 300% and back. On average, it took 7.3 s for the flow rate to stabilize,

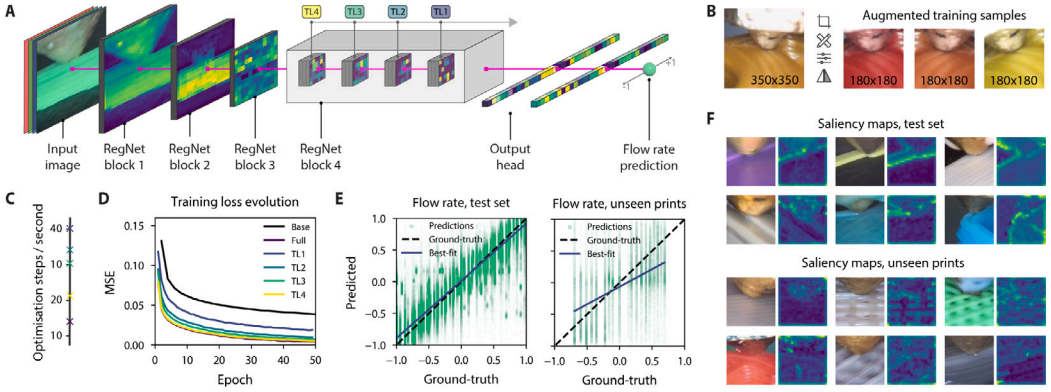


Fig. 2. Overview of the foundation model. A. Schematics and activation maps of the convolutional neural network consisting of four RegNet blocks and the regression output head. Block 4 consists of 12 sub-blocks; 3, 6, 9 and 12 of them are retrained for transfer learning schemes 1, 2, 3 and 4 respectively. B. Per-batch augmentations leading to a diverse training pool. C. Optimization speeds for the different transfer learning schemes. D. Training loss evolution during fine-tuning with different transfer learning schemes. E. Ground truth vs predicted flow rate from the foundation model for samples from the held-out test set and unseen parts. F. Selected saliency maps for samples from the held-out and unseen test sets.

with a standard deviation of 0.8 s. At 3 Hz frame rate, this equates to a window of 22 frames, which were discarded after a flow rate adjustment was triggered.

Despite the original resolution of the endoscope images being 1280×720 pixels, all remaining images were cropped to a 350×350 window centered around the location of the extrusion nozzle within the frame. This cropping procedure was implemented to diminish the data size and consequently enhance the loading speed during training. Notably, no information loss arises from cropping, as the focal point of interest corresponds to the deposition location, rendering distant pixels less significant for process monitoring.

2.2. Model training

To detect and therefore handle errors, an artificial neural network (NN) with a convolutional backbone, schematic in Fig. 2A, was trained using supervised learning to infer flow rate from nozzle-camera images via regression. The self-regulated network (RegNet) was selected as the convolutional backbone, known for its efficiency in floating-point operations per second (FLOPS) [29]. FLOPS serves as a critical metric, particularly for real-time applications such as control systems. To expedite training and ensure consistency across experiments, pre-trained weights from the original RegNet code repository were utilized for initialization. The output head of the network was adjusted to include three fully connected strata in series, each consisting of 400 neurons, 400 neurons, and 1 neuron, respectively. Optimization was guided by minimizing the mean squared error between the network's output and the labeled flow rates. The output space was projected into the natural logarithm space using Eq. (1),

$$O_{log} = log_e(O_{real}/100) \tag{1}$$

where O_{real} represents the actual flow rate and O_{log} represents the final values. This transformation mapped the 30% to 300% range between -1 and $+1$, with 100% positioned at 0. This enabled a more comprehensive and evenly spaced representation of flow rate variations.

During training, per-batch augmentations were applied to diversify and normalize the input space, as illustrated in Fig. 2B. Applied transformations included random horizontal flip, rotation in the range of -30 to 30 degrees, -10 to 10 pixel translation along the X and Y axis, 80% to 120% re-scaling and crop to random 200×200 window. Additional color jittering was applied through altering hue, brightness and contrast. The three image channels were finally normalized based on the mean and standard deviations of whole the training dataset.

For all training sessions the best batch size was found to be 128. The AdamW optimizer was selected with the weight decay parameter set to $1e-5$. AdamW decouples weight decay from the gradient update, meaning it penalises large weights and leads to better generalization performances [30]. The initial learning rate was configured at $1e-4$, complemented by a cosine annealing scheduler. The foundation model was trained using a computer with two Nvidia Quadro RTX 5000 16 GB GPUs, an Intel i9-9900K CPU, and 64 GB of RAM. For fine-tuning, iterative learning, inference, and online corrections, the same hardware configuration was employed, but GPUs were deliberately excluded by disabling their usage. This approach made experiments longer but enabled the evaluation of the efficacy of the framework in scenarios where access to high computational resources is limited (e.g. most current factory floors).

3. Results

A foundation model is trained on data from multiple build cycles and its performance is tested against seen and unseen data in Section 3.1. This foundation model is then augmented with mechanisms for uncertainty awareness and active learning in Sections 3.2 and 3.3 respectively, allowing it to progressively improve its comprehension of the task at hand. Importantly, the IL framework works well without the need for an explicit reward function, making it easier to deploy in practice when compared to reinforcement learning approaches. The generality of IL when applied to extrusion AM is demonstrated, and the experimental outcomes reveal strong performance compared to a naive fine-tuning approach, marked by optimized space usage, reduced computational load, and improved time efficiency.

3.1. Building and deploying the foundation model

A dataset comprising sixty parts, including forty 3D builds and twenty 2D builds, was compiled using the described data collection and pre-processing methods from Section 2.1. The introduction of 2D builds followed the observation of significant intra-class variability between samples acquired during the deposition of the first layer and those acquired from subsequent ones. For all parts, infill pattern, infill density, hatch orientation, and wall count, were randomized to enhance dataset diversity, while filament and nozzle orifice diameter remained consistent with PLA and 0.4 mm respectively. Occasional alterations in color and nozzle style were made for added variance. The final dataset after processing comprised $1,120,120$ images. A random sampling approach was then employed to split the dataset into training, validation, and test sets, with an 80:10:10 ratio. This allowed sufficient data for learning, tuning and assessing the network.

Using the generated dataset and model training methods detailed in Section 2.2, the foundation model was trained. After 25 training

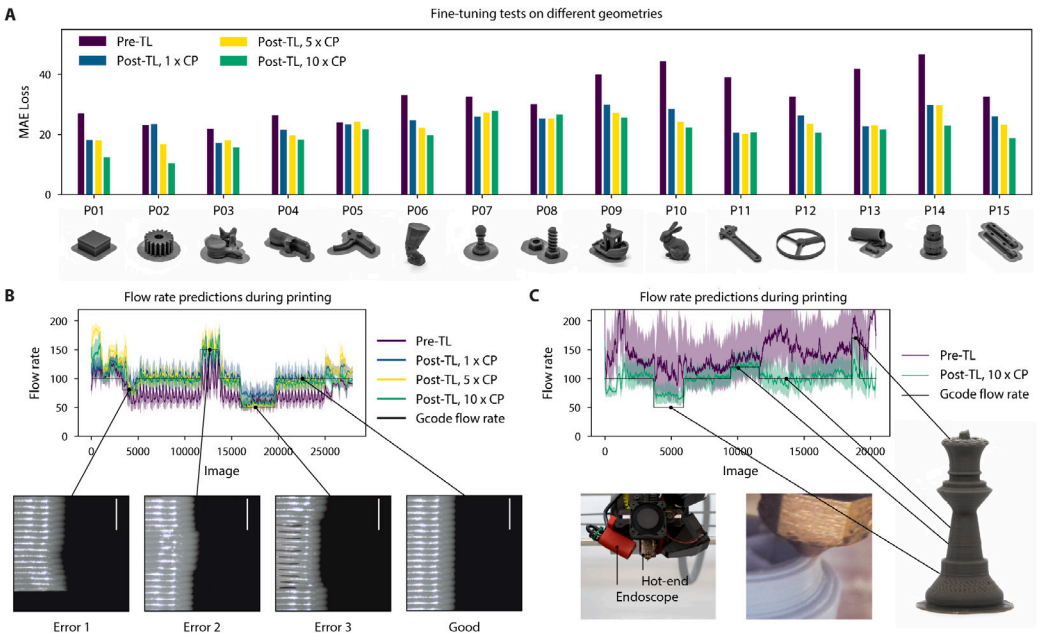


Fig. 3. Effect of fine-tuning on a specific part. A. Mean absolute error between ground truth and model predictions reported before and after fine-tuning with different amount of calibration parts (CP) for fifteen test geometries. B. Flow rate evolution during the build cycle of part P01, super-imposed with predictions from the foundation and fine-tuned models. Microscope images of the part show how over, under and normal extrusion look like on the micro scale (scale bar is 1 mm). C. Flow rate evolution during a build cycle of the Lulzbot Taz-6, super-imposed with predictions from the foundation and a fine-tuned model from 10 calibration parts. Images of the setup and sample image are also shown. Part image shows how over, under and normal extrusion look like on the part scale.

epochs, the model demonstrated convergent behavior with the MSE loss reaching 0.043 in the log-space. The mean absolute error (MAE) on the held-out test set was observed to be 12.31 in the real-space. MAE is reported as an additional metric which reflects the magnitude of the error in an interpretable manner.

The trained foundation model demonstrated consistent performance across the whole flow rate regime with good capabilities in distinguishing between over, good or under extrusion in the test set, Fig. 2E. However, a steep decline in performance was observed during inference on data from unseen parts that have been totally excluded from the training set. This shift was quantified by a deterioration to 38.24 MAE units, highlighting a significant challenge in the model's ability to generalize to new data. Saliency maps extracted using GradCAM++ [31] suggested that the performance drops may be correlated to the presence of previously unseen geometric features, causing the model to focus on abstract patterns rather than critical regions such as the most recent extrusion, Fig. 2F.

This challenge presents a pivotal concern: in practical production settings, the trained model is highly likely to encounter unseen parts. Consequently, the primary focus revolves around devising strategies to facilitate seamless adaptation to new data.

3.2. The effect of specializing in one part

A transfer learning approach was firstly employed to naively fine-tune the foundation model for recognizing features in builds that it has not previously encountered. This method allowed leveraging existing knowledge and thus providing an alternative to the traditional reliance on large, part-specific datasets. The process begins with the production of a limited number of target part samples, the "calibration parts", intentionally fabricated with specific errors as described in Section 2.1. All collected data from these parts are then concatenated into one balanced set that is used to re-train the foundation model. The re-training phase serves as a proxy for the target part, allowing the model to learn and identify task-specific feature maps. During re-training, only the final neurons are optimized with different schemes shown in Fig. 2A. Specifically, transfer learning schemes 1, 2, 3, 4 refer to optimizing 3, 6, 9, 12 of the sub-blocks in the final block of the RegNet architecture. This disables a large portion of gradient back-propagation calculations and leads to higher training speeds, Fig. 2C. However, the static neurons are not optimized, limiting their ability to adapt, and leading to higher training losses, Fig. 2D. By controlling the amount of static neurons, one can establish a good balance between fast re-training and plasticity. Transfer learning scheme 2 (6 sub-blocks re-trained) resulted in the best fit for this case and was used for the remaining experiments. The process above will be referred to as the 'calibration phase' hereafter.

The foundation model was fine-tuned on 15 different geometries, using 1, 5 and 10 calibration parts in each case, Fig. 3A. The test geometries were selected from Thingiverse to include parts with different aspect ratio and types of features (i.e., overhangs, thin walls). Additional complexity was added by slicing with unseen infill types, including some not seen during training. Grey PLA filament with 1.75 diameter (Polymaker PolyLite PLA - Grey, 1.75 mm/1000 g) was always used to ensure good comparability between different part experiments. After the calibration phase, the parts were also built with three random errors in their build cycle and their data were used for testing. A boost in the performance of the parameter predictor in tracking the three errors was observed, even with 1 calibration part, with an average decrease in MAE equal to 8.64. Similarly, using 5 or 10 calibration parts offered greater error reduction of 10.05 and 12.53 respectively. For reference, the flow rate evolution during the build cycle of a P01 part, super-imposed with predictions from different models is shown in Fig. 3B. The effects of the three errors on the macro-scale are also visualized. In general the trend observed here suggests that more calibration parts, e.g. more re-training data, lead to better improvements in terms of MAE, consistent with existing literature around transfer learning.

The foundation model was also fine-tuned on another AM system, with different specifications; a Lulzbot Taz-6 AM system equipped with a 0.6 mm nozzle and 2.85 mm acrylonitrile butadiene styrene (ABS) filament (PolyMaker PolyLite ABS - Grey, 2.85 mm/1.000 g), Fig. 3C. Similar to previous experiments, no other modifications were made to the AM system, to replicate typical operational conditions.

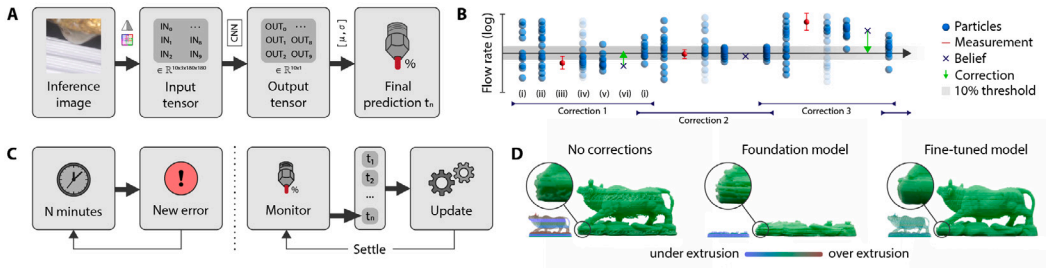


Fig. 4. Use of a probabilistic controller to correct errors. A. Schematics of the inference pipeline used to promote and simultaneously quantify robustness. B. Sample timeline of the probabilistic error correction pipeline which uses particle filters to estimate current belief. The algorithm consists of 6 sub-steps: (i) initial particle distribution, (ii) addition of adaptive particles, (iii) measurement from model, (iv) weight update using the likelihood function, (v) particle re-sampling and (vi) belief estimation and correction application. C. Schematics of the validation framework developed to stress-test the error-correction system. A random error is introduced every N minutes while closed-loop control is active and maintains the flow rate at optimal levels. D. Images and geometric flow rate diagrams of the built artefacts without, foundational and fine-tuned corrections.

Originally, the foundation model over-estimated the flow rate, with an MAE of 69.21 units. This may be due to differences between the training machine and the test machine, like the larger orifice diameter, the different nozzle geometry or the different view angle offered by the printer-specific camera mount. After fine-tuning with data from 10 calibration parts obtained through the new AM system, the MAE reduced to 19.68 demonstrating effective transfer of knowledge from one AM system to another.

The statistical significance of the fine-tuning experiments was established through multiple paired t-tests, where prior and posterior prediction errors were compared. Three different tests were used to examine how the amount of calibration parts affects the results. The null hypothesis was that the suggested framework provided no benefit whereas the alternative hypothesis was fine-tuning caused a decrease in the prediction errors with high effect. The results indicated statistically significant differences between the measurements prior and post fine-tuning, with highly acceptable p-values ($\ll 0.05$) and high test power. The observed effect size confirmed high practical significance that is proportional to the amount of calibration parts used. This supports the scalability of the framework, as the calibration phase can be tailored to the nature of the build (in terms of application) based on a trade-off between required parameter prediction accuracy and resources needed to achieve it. The full results from the statistical tests can be found in Table S1.

Enhancements in the inference pipeline, as shown in Fig. 4A, involved applying slight perturbations to each incoming image, creating ten variations V, and concatenating them into a tensor $V \times C \times W \times H$ with dimensions $\mathbb{R}^{10\times3\times180\times180}$. The network's forward pass then generated a $\mathbb{R}^{10\times1}$ tensor, with each element corresponding to each image variant. The final measurement t_n was derived by calculating the mean μ and standard deviation σ of the tensor's distribution. Utilizing μ instead of a single prediction reduced the MAE of the predictions by 3.23 on average, while σ offered insights into the robustness or uncertainty, valuable when used to control the AM system. The computational overload is minimal since the 10 images were handled by the NN in parallel.

3.3. Demonstrating an uncertainty aware controller

Applying a correction for every measurement t_n resulted in unstable control due to noise and mechanical or software delays. To overcome this challenge in control stability, multiple measurements t_n were combined at a suitable frequency to avoid vanishing or overshooting of the controlled parameter. Specifically, the parameter predictor was coupled with a probabilistic control algorithm inspired by Monte Carlo localisation (MCL), achieving autonomous error correction [32]. MCL, a technique commonly used in mobile robotics, uses a set of N particles denoted by z to represent the current state belief B for a physical value. Each particle is assigned a weight w that is directly correlated to its likelihood of being true $p(z|t_n)$. During operation, extrinsic and intrinsic measurements are used to update the likelihood of each particle, using Bayes' theorem, Eq. (2)

$$p(z|t_n) \propto e^{\frac{-(z-\mu)^2}{2\sigma^2}} \tag{2}$$

where μ and σ are the mean and uncertainty of the extrinsic measurement. At any instance, the state can be estimated as the weighted average of all surviving particles, Eq. (3)

$$B = \frac{\sum_{i=1}^{N} w_i z_i}{\sum_{i=1}^{N} w_i} \tag{3}$$

Adaptive particles are also used to deal with sudden changes in the state of the system. In general, the developed control algorithm is composed of two main phases: the update step, which includes six sub-steps shown in Fig. 4B, and the measuring step during which monitoring data are collected and processed.

The update and measuring step duration were selected to be 8 s long each, consistent with the maximum delay that may occur due to a change in flow rate. During the measuring step, the control system is allowed to collect as many measurements as possible before the update step starts. During the update step, all collected measurements are combined into one Gaussian distribution denoted as G, using weighted average and standard deviation calculations (weight being directly proportional to the robustness metric of each measurement). G is then used as the extrinsic measurement in Eq. (2) to calculate the likelihood of all existing particles and generate the posterior belief. Based on the posterior belief a correction is calculated and sent to the AM system. Estimating and forwarding the update takes a few milliseconds; during the rest of the update step the system is allowed to settle before the next measuring step starts. More details for the control algorithm can be found in supplementary material.

The validation framework in Fig. 4C, was designed as a stress test to confirm the effectiveness of the control algorithm by introducing controlled errors in the build cycle. This was achieved by switching to a random flow rate, at least 10 times during the full duration of a build cycle. To counteract the errors, a live correction system operated in parallel, striving to maintain the AM machine's flow rate as close to ideal (e.g., 100%) as possible. For consistency across various experiments, a fixed seed was utilized, ensuring the same errors were introduced in different build cycles. This validation framework was rigorously applied across different scenarios: parts corrected using both foundation and fine-tuned models, and an uncorrected sample, as illustrated in Fig. 4D. Results indicated a critical limitation in the foundation model, characterized by the vanishing flow rate due to false positive corrections. Conversely, the fine-tuned model with 5 calibration parts, demonstrated robust performance, successfully building the desired part even under the adversarial conditions imposed by the stress test. This confirms the hypothesis regarding the utility of a part-optimized parameter predictor.

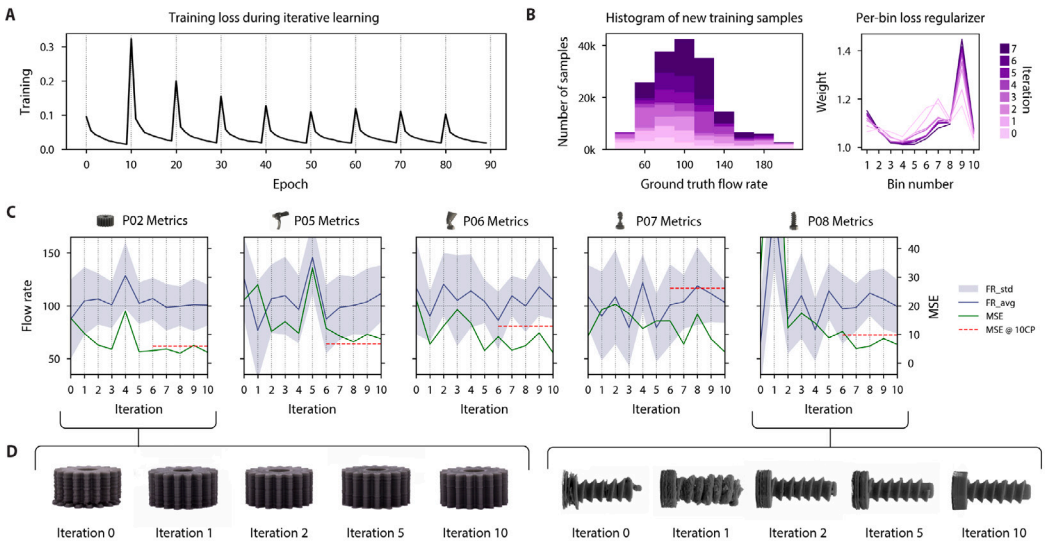


Fig. 5. Analysis of the iterative learning results. A. Example training loss plot during iterative learning. Each re-training iteration begins from the foundation model to avoid saturation points. B Histogram of the collected training samples divided in iterations. The dataset is becoming increasingly unbalanced with bias towards ideal thus a per-bin weight is used to balance the loss function. C. Performance of the iterative learning algorithm on five geometries, compared with metrics from simple transfer learning. Average flow rate and MSE deviations from ideal control plots are shown. FR_{avg}: average flow rate throughout print iteration, FR_{std}: standard deviation of the flow rate distribution throughout print iteration. MSE: mean square error between ideal and ground truth flow rate throughout print iteration. D. Images of artefacts P02, P08 stressed via the validation framework and corrected using the Iterative learning framework. Best viewed zoomed in.

3.4. A machine in the loop enables iterative learning

Mass production is a setting where processes are repeated many times. This necessitates adaptation not just across different processes but crucially, within the same process. Additionally, the repetitive nature of mass production results in a substantial influx of data, underlining the need to leverage the continuously expanding knowledge base pertinent to the target domain. Such knowledge encompasses various updates, ranging from small specification changes (like changes in nozzle style due to clogging) to new observations that are directly relevant to the task at hand. Addressing this challenge, the Iterative Learning (IL) framework was developed, detailed in Algorithm 1.

Algorithm 1 Iterative learning (IL) algorithm

Input pre-trained model θ_0, target task P
Initialize global knowledge K
while $i < i_{max}$ **do**
  Start target performance P_t
  Initialize iteration data X_t
  for (x_{inf}, y_{inf}) **while** P_t
    $\hat{y} = f(x_{inf}, \theta_t)$
    Control $control(\hat{y})$
    $X_t \parallel (x_{inf}, y_{inf})$
  $K \parallel X_t$
  Fine-tune $\theta_t = argmin_{\theta_t} L(\theta_t, K)$
end while
end

In IL, the foundational model is initially used to drive the closed-loop control of an episode of the target task. During the first episode new data are acquired and labeled via the machine-in-the-loop which knows the ground truth from firmware estimations, built-in encoders or any other form of metrology. After each iteration other than the first, new data are concatenated with all previous knowledge referred to as memory. Memory is then used to fine-tune the foundation model in preparation for the next episode. This method is intended to allow the model to learn from its own mistakes through iterative re-training during each target task episode, thereby self-correcting biases such as leaning towards predicting under or over extrusion. Specifically in the context of IL for 3D printing, this approach offers systematic advantages over the naive fine-tuning framework by eliminating the need for preparing the calibration phase and reducing the requirement for post-collection data manipulation. To efficiently manage memory, a per-bin weighting system can also be implemented, enhancing the model's accuracy and reliability, Fig. 5B.

The IL framework alongside naive transfer learning were tested on five parts also used for the experiments shown in Section 2.2. Specifically, P02, P05, P06, P07, P08 were selected to include instances where the naive transfer learning framework performed well and poorly. The results in Fig. 5C show that IL can outperform previous results in terms of MSE convergence speed, with less data, and sometimes in just 3 iterations of the target print. Final accuracy can also benefit, especially for cases where the naive transfer learning approach achieved extremely poor results like part P07.

Improvements in the average flow rate during the 3D printing process were also observed. These improvements were evident not only in stress-test conditions but also in standard operational settings, extending beyond the initial validation framework. A pivotal element of this enhancement is the notable reduction of false positive predictions. Such false positives have previously triggered unnecessary corrections, adversely affecting print quality. False positives are corrections where the belief may trigger a correction in the wrong direction. By minimizing these, IL avoided counterproductive interventions, particularly important in typical printing scenarios where errors are relatively rare instances.

In Fig. 5D images of the printed artefacts are provided, specifically P02 and P08, to illustrate the progressive refinement in part precision across iterations. In the case of P08, during the very first iteration, the foundation model exhibited bias towards predicting over-extrusion and caused the actual flow rate below ideal at 70.13% ± 12.21 on average. After iteration 1 the framework learned from this mistake, but overcompensated for it, resulting in heavy over-extrusion with average flow rate equal to 182.31%±9.20. By iteration 2, the framework demonstrated its adaptive learning capability, striking a balance between the two extremes with an improved and more consistent average flow rate of 91.22% ± 6.32. By this iteration the controller is able to build the part as-designed, with the model effectively correcting its

Table 1
Ablating model size and framework components. Evaluating impact of foundation model size and engineering decisions on downstream tasks. *Parameters* refers to trainable variables of the neural network architecture. *Latency* refers to average inference time across 10 samples, reported in ms per frame on a CPU.

Architecture	Parameters	Latency	MAE
Full pipeline with different architectures			
RegNetX_400MF	2.1 M	**9.34**	19.21
RegNetX_1_6GF	2.6 M	15.87	18.55
RegNetX_8GF	3.7 M	42.41	17.42
RegNetX_32GF	6.2 M	140.35	**17.01**
Ablation study on RegNetX_400MF			
⊖ domain-specific pre-training			24.43
⊖ uncertainty awareness			26.21
⊖ iterative learning			29.20

initial biases in two shots. The process continued through subsequent iterations with the memory constantly expanding, and by iteration 10 the printed artefact showcased a high degree of precision and geometric fidelity with 101.20%±5.55 average flow rate. Similar behavior has been observed for other parts, including the one seen in Fig. 1.

3.5. Ablating the framework components

Model size. A model size ablation was done by training various RegNet architectures, ranging from RegNetX_400MF to RegNetX_32GF, Table 1. This analysis revealed a clear trade-off between model complexity and performance. Following transfer learning scheme 2, as the number of trainable parameters increased from 2.1 millions to 6.2 millions, a consistent decrease in final MAE was observed, from 19.21 to 17.01, indicating enhanced predictive accuracy. Concurrently, a substantial increase in computational latency was recorded, for both training and inference. Specifically, the smallest model can achieve inference in 9.34 ms, in stark contrast to the 140.35 ms needed from the largest one. These findings further support the need for larger models only if and when the computational resources to support them are available.

Framework components. Further ablation on the smallest model, RegNetX_400MF, reveals the significance of the main framework components; pre-training the foundation model with domain-specific data, uncertainty modeling using the Monte Carlo inference pipeline, and using IL instead of naive transfer learning. All three components enhance model performance, as their removal resulted in a marked increase in MAE.

4. Discussion and open questions

In this work downstream task optimization of a foundation model was combined with uncertainty awareness and active learning with a machine in the loop to enable better error detection and correction specifically for higher production volume AM. The practical significance of the proposed approach has been supported using statistical tests. For AM to reach its full potential, it needs to become more viable in higher-volume applications. The presented experiments demonstrated potential to enhance quality monitoring and continually improving control in high-volume AM. This could subsequently improve the productivity and sustainability of AM via better quality (in terms of precision and repeatability) parts and reduced scrap. In turn, also enabling novel custom products and distributed manufacturing systems across areas such as medical devices and the aerospace industry. Such vision-based methods may be more challenging to use with transparent materials, yet alternative imaging sensors, such as infrared cameras, can provide assistance, particularly in scenarios where heat is involved. The methodology developed could also find uses in other manufacturing processes, where a part-optimized system can enable the mass production of a specific part with increased quality.

From a control perspective, the IL framework can be seen as combining aspects of active learning and iterative learning control (ILC) [33]. Historically, ILC research has predominantly concentrated on refining control mechanisms using a predefined set of signals, but the potential of task iterations as a means of enhancing sensor-derived information remains underexplored. ILC's integration with deep learning is also limited. On the same note, current active learning methodologies primarily depend on human intervention for labeling ambiguous samples and typically do not account for the potential of iterative enhancement in specific tasks. This work addresses these critical gaps, presenting a unified approach that combines the iterative elements of ILC with deep learning and the dynamic learning capabilities of active learning. This integration might hold significant potential for broader applications across various fields.

In future, multiple networks trained during the IL algorithm could be used as an ensemble to improve the uncertainty awareness of the framework. Similarly, the active learning part of the framework can be improved in terms of space complexity, if the necessity to explicitly store memory was to be mitigated. The convergence properties of the IL framework, likely to be similar to those of semi-supervised learning, may also be investigated.

CRediT authorship contribution statement

Christos Margadji: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Douglas A.J. Brion:** Investigation, Conceptualization. **Sebastian W. Pattinson:** Writing – review & editing, Supervision, Resources, Project administration, Funding acquisition, Conceptualization.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Sebastian Pattinson reports financial support was provided by UK Research and Innovation. Christos Margadji reports financial support was provided by UK Research and Innovation. Douglas Brion reports financial support was provided by UK Research and Innovation. Sebastian Pattinson reports a relationship with Matta Labs that includes: board membership and equity or stocks. Douglas Brion reports a relationship with Matta Labs that includes: board membership, employment, and equity or stocks. Sebastian Pattinson has patent pending to Matta Labs. Douglas Brion has patent pending to Matta Labs. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

This work has been funded by the Engineering and Physical Sciences Research Council (EPSRC) Ph.D. Studentship, United Kingdom EP/N509620/1 to CM and DAJB, Engineering and Physical Sciences Research Council, United Kingdom award EP/V062123/1 to SWP. For the purpose of open access, the author has applied a Creative Commons Attribution (CC-BY) license to any Author Accepted Manuscript version arising.

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.addma.2024.104271.

References

[1] T.D. Ngo, A. Kashani, G. Imbalzano, K.T. Nguyen, D. Hui, Additive manufacturing (3D printing): A review of materials, methods, applications and challenges, Composites B 143 (2018) 172–196, http://dx.doi.org/10.1016/j.compositesb.2018.02.012, URL https://www.sciencedirect.com/science/article/pii/S1359836817342944.

[2] G. Haghiashtiani, K. Qiu, J.D.Z. Sanchez, Z.J. Fuenning, P. Nair, S.E. Ahlberg, P.A. Iaizzo, M.C. McAlpine, 3D printed patient-specific aortic root models with internal sensors for minimally invasive applications, Sci. Adv. 6 (35) (2020) eabb4641, http://dx.doi.org/10.1126/sciadv.abb4641, URL https://www.science.org/doi/abs/10.1126/sciadv.abb4641.

[3] J.C. Najmon, S. Raeisi, A. Tovar, Review of additive manufacturing technologies and applications in the aerospace industry, in: F. Froes, R. Boyer (Eds.), Additive Manufacturing for the Aerospace Industry, Elsevier, 2019, pp. 7–31, http://dx.doi.org/10.1016/B978-0-12-814062-8.00002-9, URL https://www.sciencedirect.com/science/article/pii/B9780128140628000029.

[4] I. Hager, A. Golonka, R. Putanowicz, 3D printing of buildings and building components as the future of sustainable construction? Procedia Eng. 151 (2016) 292–299, http://dx.doi.org/10.1016/j.proeng.2016.07.357, URL https://www.sciencedirect.com/science/article/pii/S1877705816317453, Ecology and new building materials and products 2016.

[5] M. Baechle-Clayton, E. Loos, M. Taheri, H. Taheri, Failures and flaws in fused deposition modeling (FDM) additively manufactured polymers and composites, J. Compos. Sci. 6 (7) (2022) http://dx.doi.org/10.3390/jcs6070202, URL https://www.mdpi.com/2504-477X/6/7/202.

[6] O.A. Mohamed, S.H. Masood, J.L. Bhowmik, Optimization of fused deposition modeling process parameters: a review of current research and future prospects, Adv. Manuf. 3 (1) (2015) 42–53, http://dx.doi.org/10.1007/s40436-014-0097-7.

[7] H.-Q. Xu, J.-C. Liu, Z.-Y. Zhang, C.-X. Xu, A review on cell damage, viability, and functionality during 3D bioprinting, Milit. Med. Res. 9 (1) (2022) 70, http://dx.doi.org/10.1186/s40779-022-00429-5.

[8] Y. Xu, H. Zhang, B. Šavija, S. Chaves Figueiredo, E. Schlangen, Deformation and fracture of 3D printed disordered lattice materials: Experiments and modeling, Mater. Des. 162 (2019) 143–153, http://dx.doi.org/10.1016/j.matdes.2018.11.047, URL https://www.sciencedirect.com/science/article/pii/S0264127518308530.

[9] S.J. Schuldt, J.A. Jagoda, A.J. Hoisington, J.D. Delorit, A systematic review and analysis of the viability of 3D-printed construction in remote environments, Autom. Constr. 125 (2021) 103642, http://dx.doi.org/10.1016/j.autcon.2021.103642, URL https://www.sciencedirect.com/science/article/pii/S0926580521000935.

[10] Y. Tlegenov, G.S. Hong, W.F. Lu, Nozzle condition monitoring in 3D printing, Robot. Comput.-Integr. Manuf. 54 (2018) 45–55, http://dx.doi.org/10.1016/j.rcim.2018.05.010, URL https://www.sciencedirect.com/science/article/pii/S073658451730443X.

[11] C. Kim, D. Espalin, A. Cuaron, M.A. Perez, E. MacDonald, R.B. Wicker, A study to detect a material deposition status in fused deposition modeling technology, in: 2015 IEEE International Conference on Advanced Intelligent Mechatronics, AIM, 2015, pp. 779–783, http://dx.doi.org/10.1109/AIM.2015.7222632.

[12] J. Guo, J. Wu, Z. Sun, J. Long, S. Zhang, Fault diagnosis of delta 3D printers using transfer support vector machine with attitude signals, IEEE Access 7 (2019) 40359–40368, http://dx.doi.org/10.1109/ACCESS.2019.2905264.

[13] S. Zhang, Z. Sun, C. Li, D. Cabrera, J. Long, Y. Bai, Deep hybrid state network with feature reinforcement for intelligent fault diagnosis of delta 3-D printers, IEEE Trans. Ind. Inform. 16 (2) (2020) 779–789, http://dx.doi.org/10.1109/TII.2019.2920661.

[14] P.K. Rao, J.P. Liu, D. Roberson, Z.J. Kong, Sensor-Based Online Process Fault Detection in Additive Manufacturing, in: International Manufacturing Science and Engineering Conference, Vol. 2: Materials; Biomanufacturing; Properties, Applications and Systems; Sustainable Manufacturing, 2015, V002T04A010, http://dx.doi.org/10.1115/MSEC2015-9389, arXiv:https://asmedigitalcollection.asme.org/MSEC/proceedings-pdf/MSEC2015/56833/V002T04A010/4424804/v002t04a010-msec2015-9389.pdf.

[15] H. Wu, Y. Wang, Z. Yu, In situ monitoring of FDM machine condition via acoustic emission, Int. J. Adv. Manuf. Technol. 84 (5) (2016) 1483–1495, http://dx.doi.org/10.1007/s00170-015-7809-4.

[16] K.T. Estelle, B.A. Gozen, Precision flow rate control during micro-scale material extrusion by iterative learning of pressure-flow rate relationships, Addit. Manuf. 82 (2024) 104031, http://dx.doi.org/10.1016/j.addma.2024.104031, URL https://www.sciencedirect.com/science/article/pii/S2214860424000770.

[17] J. Straub, Initial work on the characterization of additive manufacturing (3D printing) using software image analysis, Machines 3 (2) (2015) 55–71, http://dx.doi.org/10.3390/machines3020055, URL https://www.mdpi.com/2075-1702/3/2/55.

[18] K. He, Q. Zhang, Y. Hong, Profile monitoring based quality control method for fused deposition modeling process, J. Intell. Manuf. 30 (2) (2019) 947–958, http://dx.doi.org/10.1007/s10845-018-1424-9.

[19] T. Huang, S. Wang, S. Yang, W. Dai, Statistical process monitoring in a specified period for the image data of fused deposition modeling parts with consistent layers, J. Intell. Manuf. 32 (8) (2021) 2181–2196, http://dx.doi.org/10.1007/s10845-020-01628-4.

[20] O. Holzmond, X. Li, In situ real time defect detection of 3D printed parts, Addit. Manuf. 17 (2017) 135–142, http://dx.doi.org/10.1016/j.addma.2017.08.003, URL https://www.sciencedirect.com/science/article/pii/S2214860417301100.

[21] F.G. Cunha, T.G. Santos, J. Xavier, In situ monitoring of additive manufacturing using digital image correlation: A review, Materials 14 (6) (2021) http://dx.doi.org/10.3390/ma14061511, URL https://www.mdpi.com/1996-1944/14/6/1511.

[22] Z. Zhang, I. Fidan, M. Allen, Detection of material extrusion in-process failures via deep learning, Inventions 5 (3) (2020) http://dx.doi.org/10.3390/inventions5030025, URL https://www.mdpi.com/2411-5134/5/3/25.

[23] Z. Jin, Z. Zhang, G.X. Gu, Automated real-time detection and prediction of interlayer imperfections in additive manufacturing processes using artificial intelligence, Adv. Intell. Syst. 2 (1) (2020) 1900130, http://dx.doi.org/10.1002/aisy.201900130, arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/aisy.201900130 URL https://onlinelibrary.wiley.com/doi/abs/10.1002/aisy.201900130.

[24] D.A.J. Brion, S.W. Pattinson, Generalisable 3D printing error detection and correction via multi-head neural networks, Nature Commun. 13 (1) (2022) 4654, http://dx.doi.org/10.1038/s41467-022-31985-y.

[25] D.A.J. Brion, S.W. Pattinson, Quantitative and real-time control of 3D printing material flow through deep learning, Adv. Intell. Syst. 4 (11) (2022) 2200153, http://dx.doi.org/10.1002/aisy.202200153, URL https://onlinelibrary.wiley.com/doi/abs/10.1002/aisy.202200153.

[26] M.V. Johnson, K. Garanger, J.O. Hardin, J.D. Berrigan, E. Feron, S.R. Kalidindi, A generalizable artificial intelligence tool for identification and correction of self-supporting structures in additive manufacturing processes, Addit. Manuf. 46 (2021) 102191, http://dx.doi.org/10.1016/j.addma.2021.102191, URL https://www.sciencedirect.com/science/article/pii/S2214860421003535.

[27] J.M. Gardner, K.A. Hunt, A.B. Ebel, E.S. Rose, S.C. Zylich, B.D. Jensen, K.E. Wise, E.J. Siochi, G. Sauti, Machines as craftsmen: Localized parameter setting optimization for fused filament fabrication 3D printing, Adv. Mater. Technol. 4 (3) (2019) 1800653, http://dx.doi.org/10.1002/admt.201800653, arXiv:https://onlinelibrary.wiley.com/doi/pdf10.1002/admt.201800653 URL https://onlinelibrary.wiley.com/doi/abs/10.1002/admt.201800653.

[28] R. Bommasani, D.A. Hudson, E. Adeli, R. Altman, S. Arora, S. von Arx, M.S. Bernstein, J. Bohg, A. Bosselut, E. Brunskill, E. Brynjolfsson, S. Buch, D. Card, R. Castellon, N. Chatterji, A. Chen, K. Creel, J.Q. Davis, D. Demszky, C. Donahue, M. Doumbouya, E. Durmus, S. Ermon, J. Etchemendy, K. Ethayarajh, L. Fei-Fei, C. Finn, T. Gale, L. Gillespie, K. Goel, N. Goodman, S. Grossman, N. Guha, T. Hashimoto, P. Henderson, J. Hewitt, D.E. Ho, J. Hong, K. Hsu, J. Huang, T. Icard, S. Jain, D. Jurafsky, P. Kalluri, S. Karamcheti, G. Keeling, F. Khani, O. Khattab, P.W. Koh, M. Krass, R. Krishna, R. Kuditipudi, A. Kumar, F. Ladhak, M. Lee, T. Lee, J. Leskovec, I. Levent, X.L. Li, X. Li, T. Ma, A. Malik, C.D. Manning, S. Mirchandani, E. Mitchell, Z. Munyikwa, S. Nair, A. Narayan, D. Narayanan, B. Newman, A. Nie, J.C. Niebles, H. Nilforoshan, J. Nyarko, G. Ogut, L. Orr, I. Papadimitriou, J.S. Park, C. Piech, E. Portelance, C. Potts, A. Raghunathan, R. Reich, H. Ren, F. Rong, Y. Roohani, C. Ruiz, J. Ryan, C. Ré, D. Sadigh, S. Sagawa, K. Santhanam, A. Shih, K. Srinivasan, A. Tamkin, R. Taori, A.W. Thomas, F. Tramèr, R.E. Wang, W. Wang, B. Wu, J. Wu, Y. Wu, S.M. Xie, M. Yasunaga, J. You, M. Zaharia, M. Zhang, T. Zhang, X. Zhang, Y. Zhang, L. Zheng, K. Zhou, P. Liang, On the opportunities and risks of foundation models, 2022, arXiv:2108.07258.

[29] J. Xu, Y. Pan, X. Pan, S. Hoi, Z. Yi, Z. Xu, RegNet: Self-regulated network for image classification, 2021, arXiv:2101.00590.

[30] I. Loshchilov, F. Hutter, Decoupled weight decay regularization, 2019, arXiv:1711.05101.

[31] A. Chattopadhay, A. Sarkar, P. Howlader, V.N. Balasubramanian, Grad-CAM++: Generalized gradient-based visual explanations for deep convolutional networks, in: 2018 IEEE Winter Conference on Applications of Computer Vision, WACV, IEEE, 2018, http://dx.doi.org/10.1109/wacv.2018.00097.

[32] F. Dellaert, D. Fox, W. Burgard, S. Thrun, Monte Carlo localization for mobile robots, in: Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No.99CH36288C), Vol. 2, 1999, pp. 1322–1328, http://dx.doi.org/10.1109/ROBOT.1999.772544.

[33] D. Owens, J. Hätönen, Iterative learning control — An optimization paradigm, Annu. Rev. Control 29 (1) (2005) 57–70, http://dx.doi.org/10.1016/j.arcontrol.2005.01.003, URL https://www.sciencedirect.com/science/article/pii/S1367578805000106.