Journal of Manufacturing Processes 71 (2021) 743–752

Contents lists available at ScienceDirect

Journal of Manufacturing Processes

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

Joint active search and neuromorphic computing for efficient data exploitation and monitoring in additive manufacturing

Ruimin Chen [a], Mohsen Imani [b], Farhad Imani [c,*]

[a] Independant Researcher, Tolland, CT, 06084
[b] Department of Computer Science, University of California Irvine, Irvine, CA 92697, USA
[c] Department of Mechanical Engineering, University of Connecticut, Storrs, CT 06269, USA

ARTICLE INFO

Keywords:
Active search
Additive manufacturing
Brian-inspired computing
Graph modeling
Bandit problem
Data annotation

ABSTRACT

The recent integration of imaging technology with additive manufacturing (AM) leads to the plethora of in-process and high-dimensional data. Machine learning (ML) methods have been implemented to improve understanding of defect formation in AM-built parts and controlling process variability in real-time. However, modern ML methods, in particular deep neural networks, are empowered by massive high-quality labeled data, which are limited in AM due to the following reasons: First, large data labeling is often tedious, costly, and requires substantial human efforts with considerable expertise. Second, the performance of the learning methods depends to a great extent on the presence of positive data instances (i.e., defective) as they are more informative for monitoring. Third, the rare positives result in a severe imbalanced dataset poses critical challenges in training ML methods designed with the assumption that the input contains an equal number of instances from each class. In this research, we propose novel annotation and learning with limited number of data through the integration of active search and hyperdimensional computing (HDC). The active search is developed to benefit from a single bandit model to learn about the data distribution (exploration) while sampling from the regions potentially containing more positives (exploitation). HDC is introduced as an alternative computing method that mimics important brain functionalities and encodes data with high-dimensional vectors, thereby enabling single-pass learning with just a few samples. Experimental results on a real-world case study of drag link joint build show the proposed model locates the rare positives thoroughly and detects lack of fusion defects with the accuracy of 89.58%, in 3.221 ± 0.029 second training time and with only 66 sample data. The joint active search and neuromorphic computing framework is shown to have strong potentials for general applications in a diverse set of domains with in-situ imaging data.

1. Introduction

The rapid emergence of technology has led to the shift from sparse and episodic measurement to in-situ and high-dimensional optical imaging. As such, high-resolution data become readily available, which offer unprecedented opportunity to address the ever-increasing complexity of systems and control the process variations in various domains, including healthcare [1–3] and manufacturing [4–6]. In particular, in-situ imaging is recently integrated with additive manufacturing (AM) to empower the next generation of processes and tackle inconsistency issues (e.g., variations in geometrical accuracy, process stability, and mechanical properties). This provides a possibility to detect the onset of AM defects (e.g., porosity, crack, delimitation, and lack of fusion) prior to completion of the build. Therefore, it prevents defects permanently sealed in by subsequent layers, which may impact the build's strength, residual stress, hardness, and fatigue life.

Supervised machine learning (ML) methods are recently invested in AM to establish a framework that seamlessly integrates in-situ image-based monitoring with compensation methods to effectively detect and subsequently correct the process drift and anomalies toward high-quality printing [7,8]. However, the capability of these methods is highly dependent on the availability of labeled datasets that meet certain technical goals for training and validation. Although a remarkable amount of data can be generated easily with the cameras and connected devices in AM, three key challenges hinder the development of state-of-the-art supervised learning methods:

* Corresponding author.
E-mail address: farhad.imani@uconn.edu (F. Imani).

https://doi.org/10.1016/j.jmapro.2021.09.048
Received 7 August 2021; Received in revised form 26 September 2021; Accepted 28 September 2021
Available online 19 October 2021

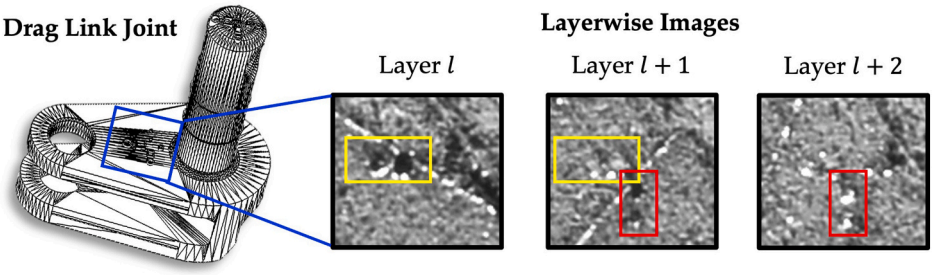

Fig. 1. An example of data annotation with the information of sequential structure. Part design is shown in the left, and layerwise images of three consecutive layers are presented in the right. Yellow and red boxes indicate intentional defects (represent lack of fusion) in the drag link build. The defects in yellow boxes of layer $l+1$ are hard to confirm without the information from layer l. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

- *Annotation cost*: While labeling datasets is quietly affordable for social networks, consumer preferences, and finance, such task is usually expensive in terms of the consumed time, cost, and human labor [9]. In other words, rather than normal individuals, costly human expertise is required to differentiate the presence of various types of defects from images.
- *Data velocity*: The next generation of AM enables advanced product designs and capabilities but increasingly relies on highly industrial sensing and control systems. A typical smart factory can generate around 5 petabytes (i.e., 5 million gigabytes) per week [10]. However, only a small number of data can be utilized for supervised learning due to the lack of labels.
- *Limited number of rare positives*: Positive data (i.e., data with defects) are more informative for learning in ML applications; however, it is arduous to locate the rare positives. In addition, severe imbalances pose challenges for many predictive models due to the assumption of equality in the number of instances for each class in input data. Although data augmentation techniques (e.g., crop, flip, rotation, scale, and translation) may have an explicit regularization effect, their explotation leads to the ineffcient learning, resulting in poor prediction capability.

Therefore, the contribution of this research is two-fold: (1) we propose new query strategies to find the rare positives, and (2) we develop new supervised learning methodologies to learn useful information within a limited number of data. Our solution utilizes the idea of active search and neuromorphic computing to swiftly train with minimal data for supervised learning. We first leverage active search on a graph to actively find rare positives as many as possible within a given budget. Such searching problems often face an exploration and exploitation dilemma — we need to locate the rare positives and explore the distribution of the dataset by selecting the negatives at the same time.

Because materials are deposited and then sintered or melted in the layer-by-layer fashion in AM, there exist temporal (i.e., across layers) correlations in layerwise images. The solidarity process impacts the integrity of subsequent layers. Such dynamics are critical to gaining an in-depth understanding of defect formation and propagation. Therefore, label of an image can be inferred from other images with sequential structure because parts are fabricated in a layer-by-layer fashion. For example, intentional flaws were embedded in the build at four different locations along the build-up direction intersected with cubical and cylindrical patterns in the design of the drag link joint build. As shown in Fig. 1, flaws can be seen more clearly from the optical image when it first shows (see yellow box in layer l). However, powder from the next layer might fill the flaw (see yellow box in layer $l+1$) and interfere with the annotator's decision making. Information from other layers can support the decision making in the current layer due to the sequential relationship between layers. A similar condition can be seen from the example in the red boxes. As such, having the sequential information from the previous layer can help the annotator label the image correctly and detect semantic and logic errors. Therefore, we leverage graph bandit to balance the exploration and exploitation in active search and at the same time address the data dependencies with the network structure.

Further, inspired by human memory, we systematically design hyperdimensional computing (HDC) primitives to represent manufacturing data in high-dimensional space for supporting highly accurate training with a limited number of data. We propose the HDC, an adaptive training framework for accurate, efficient, and robust learning. HDC is motivated by the observation that the human brain operates on high-dimensional data representations. Objects are thereby encoded with high-dimensional vectors, called *hypervectors*, which have thousands of elements [11,12]. HDC incorporates learning capability along with typical memory functions of storing/loading information and mimics several important functionalities of the human memory model with vector operations, which are computationally tractable and mathematically rigorous in describing human cognition. As a result, HDC has the capability to expose hidden features, enabling single-pass learning with just a few samples.

The remainder of this paper is organized as follows: Section 2 provides a literature review on the relevant methods of active learning, active search, and neuromorphic computing. Section 3 presents the proposed methodology. Experimental design and results on the drag link joint part based on the real-world case study are given in Section 4. In the end, in Section 5, we conclude this paper by highlighting gaps of existing learning methods in AM and then provide an overview of the proposed methodology.

2. Research background

2.1. Additive manufacturing process dynamics and machine learning

In the domain of AM, one of the major challenges is the occurrence of various defects such as cracks, delamination, distortion, rough surface, lack of fusion, porosity, foreign inclusions, and process instability [13–16]. Area with limited pathways for heat transfer can manifest as regions of high residual stress generating cracks, delamination, and distortion. Porosity-induced part failure can occur due to lack of fusion and keyholing. Surface tension causes balling of the melt pool, which solidifies into rough surfaces that encourage crack initiation and growth [17]. AM defects usually originate from the layerwise material deposition process. Some may also propagate from the previous layer to subsequent layers, further causing the failure of the entire build. The current quality assurance in AM embodies design of experiment and simulation to reveal the underlying mechanism for the formation of specific features during fabrication (e.g., melt pool geometry, keyhole, and microstructure). However, trial-and-errors and simulation methods (e.g., finite element analysis and computational fluid dynamics) are often expensive and time-consuming. Therefore, many researchers have explored the feasibility of introducing ML methods to solve the quality challenges in AM. A significant amount of data, mainly visual images, are collected and processed to train different ML algorithms to monitor the printing process.

In practice, domain experts annotate experimental data through comparison with X-ray computed tomography (XCT) scans for supervised learning tasks. With the labeled dataset, ML methods learn from

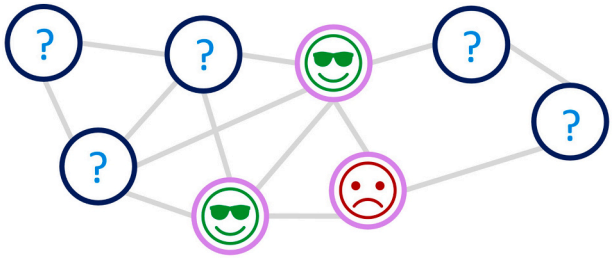

Fig. 2. An example of active search on graph. We need to determine which vertex to query next based on the observed vertices in magenta. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

the training set and make inference, perform predictions, and determine optimal process parameters. Gaussian process (GP) [18], conventional learning (e.g., support vector machine and k-nearest neighbor) [19,20], and neural network [21] are three major models. For example, Tapia et al. [22] developed a GP-based surrogate model to construct 3D response maps of melt pool depth versus process parameters. Wang et al. [23] leveraged the support vector machine to quantify the top build surface condition narrow down the process window for the electron beam melting (EBM) process. Scime and Beuth performed k-means clustering [24] and multi-scale CNN [25] to train the system to correctly classify the powder-bed image patches into seven types, based on the images captured during the powder bed fusion process. However, lack of enough training data, computational complexity, and overfitting proneness are the main drawbacks of the current ML practices in AM. The acquisition of a sufficient amount of data representative of signal patterns in the presence of defects or faulty states may be difficult to have in practice, and data labeling may be a time-consuming and troublesome task [26]. Furthermore, large AM datasets are limited owing to the infinite number of combinations in build design, material, and platform (i.e., one model for each single system, not generalizable to other systems, even similar ones).

2.2. Active search

Active search is an increasingly pivotal learning problem in which we use a limited budget of label queries to discover as many members of a certain class as possible [27]. It is a special realization of active learning (also named query learning or optimal experimental design) that annotates data for a specific model by evaluating the performance on labeled data. Active search is studied in many real-world applications, such as drug discovery [28], sentiment analysis [29], and recommendation systems [30], just to name a few.

The labels of data are not known at the beginning of the search but can be revealed by querying the annotator during the annotation process. Note that the annotator is also named as oracle in the literature. Therefore, the goal of the active search is to design a query strategy that is able to sequentially query points to find are positives (i.e., valuable points) under a labeling budget in an efficient manner. Often, this query strategy faces the fundamental dilemma between exploration and exploitation, whether to search for new regions of valuable points (exploration) or take advantage of the currently most-promising regions (exploitation) [31]. Bandit algorithms [32,33] are widely utilized to manage this trade-off in active search because selecting an item can be related to pulling an arm. For example, Antos et al. [34] focused on multi-armed bandits with budget constraints, the arm-pull is costly, and their model is limited by a fixed budget. Zhu et al. [35] proposed a Meta-Bandits to evaluate a set of base bandits and annotate the rare positives efficiently.

As big data poses new challenges for models to deal with complex dependencies, there have been some attentions to active search in the graph setting. Wang et al. [27] proposed a myopic method for active search on graphs. Vanchinathan et al. [36] designed a Gaussian process (GP) based algorithm to extend the upper-confidence bound (GP-UCB) algorithm [37] for stochastic optimization with theoretical guarantees. Ji and Han [38] integrated *V*-optimality in their experimental design, and Ma et al. [39,40] further improve the state-of-the-art by combining GP-UCB with Σ-optimality. In this research, we leverage the graph to cope with the complex structure in manufacturing data. Here, data are represented as vertices, and their relationships are encoded by edges. As shown in Fig. 2, the vertices in magenta color are the observed nodes where we have both rare positives (i.e., happy face) and negatives (i.e., sad face). We need to determine which vertex to query next based on the observed vertices through active search.

Recently, several works have recognized the opportunities of introducing active learning to aid data labeling and annotation in manufacturing. Rožanec et al. [41] discussed how to leverage active learning to identify missing knowledge in the learning of manufacturing ontology. Similarly, Zajec et al. [42] proposed an active learning-based module that receives input from the database and knowledge graph to suggest data instances that are expected to be most informative to the system. Instead of recognizing the missing information, our objective of active search is to identify the rare positives with as few iterations as possible. Wuest et al. [43] mentioned that active learning could be used for obtaining labeled training data for manufacturing problems. Meng et al. [9] argued the missing functionality of active learning in the AM field. Very limited work has been done to study the performance of active learning in the field of AM. To the best of our knowledge, this work is one of the first works to investigate the active search in the field of manufacturing.

2.3. Neuromorphic computing

HDC, also referred to as brain-inspired or neuromorphic computing, is commonly used for representations of structured knowledge and computer-based semantic reasoning. HDC is a new ML paradigm that can transform data into knowledge at a very low cost and with comparable accuracy to state-of-the-art methods for diverse applications. It is also robust to errors in communication - it can correctly infer even when as much as 30% of the bits are corrupted [44,11]. There are two main components in HDC, encoding and arithmetic operations. HDC encodes raw data into high-dimensional vectors (hypervectors), which can have as many as 10,000 bits. Processing data requires simple arithmetic operations, rotation, and nearest neighbor search, all of which are easy to accelerate in hardware [45–47], thus leading to very low-power and high-performance implementations.

The development of HDC was ignited by studies on brain activity showing that the processing of even simple mental events requires concurrent operation in many separated neurons [48]. Inspired by this observation, information in HDC is similarly represented in this fashion: a single concept is associated with a pattern of activation of many neurons. This is achieved by using a vector with very large dimensions. Several different types of HDC have been introduced, each using different representations [49,50]. HDC-based learning reduces the communication costs since training is error-tolerant against data noise and converges within a few epochs to achieve comparable accuracy to the state-of-the-art methods [11,51–53], while requiring a very small amount of data. As a result, recent years show a growing interest in implementation of HDC to sensory data in various systems. For example, Emruli et al. [54] modeled dependencies in temporal sequences of heterogeneous measurements using HDC. Imani et al. [11] and Khaleghi [11] proposed HDC for enabling lightweight privacy and security. Other applications of HDC include natural gesture recognition [55] and modality classification of medical images [56]. These results provide empirical evidence that HDC has the potential to become a powerful tool for the analysis of complex dependencies between image data in additive manufacturing.

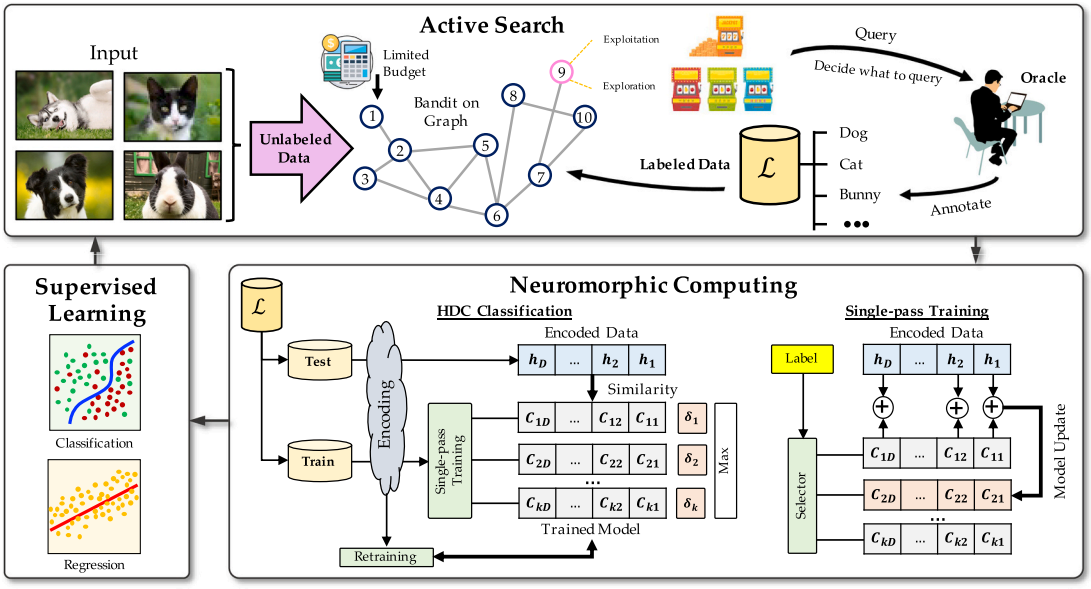

Fig. 3. The flowchart of the proposed methodology. We integrate the active search and neuromorphic computing for efficient supervised learning in advanced manufacturing.

3. Methodology

In this paper, we combine active search and neuromorphic computing for the rare positives in additive manufacturing. As shown in Fig. 3, the proposed methodology contains two main steps. First, we perform the active search on a graph to find all rare positives with a budget efficiently. We relate the problem to multi-arm bandit (MAB) and leverage the graph-UCB to address the exploration-exploitation dilemma. Note that here, we do not allow data replacement for the active search. Then, we conduct neuromorphic computing for supervised learning with a limited number of training and testing data.

3.1. Active search and bandits on graph for data annotation

Given a set of N vertices $\mathscr{V} = \{v_1, v_2 \ldots, v_N\}$ with their labels $\mathbf{L} = \{l_1, l_2, \ldots, l_N\}$ where $l_i \in \{0, 1\}$, we aim to preform active search in a graph with a given structure $\mathscr{G} = \{\mathscr{V}, \mathscr{E}\}$ and identify all vertices with l_i=1. Here, if there is a relationship between v_i and v_j, an undirected edge (v_i, v_j) is added to $\mathscr{E}$.

Let $O_T = \{v_1^*, \ldots v_T^*\}$ denotes the optimal sequence with the budget of T. Our objective is to design a query strategy that generates a query sequence $S_T = \{v_1, \ldots v_T\}$ which minimizes the regret between the proposed strategy and the optimal strategy

$$\text{Regret}(T) = \sum_{t=1}^{T} r\left(v_t^*\right) - \mathrm{E}\left[\sum_{t=1}^{T} r(v_t)\right] \tag{1}$$

where $r(v)$ is the function that predicts the reward of vertex v_i when it is selected for annotation. Note that sequences O_T and S_T do not allow repeated elements. Our model assumes that there is one annotator who does not make mistake in this work. As such, our objective for the active search here is to find the rare positives with the minimum number of iterations.

Denote the vector of rewards of all nodes in $\mathscr{G}$ as $\mathbf{v} \in \mathbb{R}^N$, the joint distribution of these random variables can be written as

$$logp(\mathbf{r}) \sim -\sum_{i=1}^{N}\sum_{j=1}^{N}\frac{A_{i,j}\left(r_i - r_j\right)^2}{2} - \sum_{j=1}^{N}\frac{\omega\left(r_j - \mu_0\right)}{2} \tag{2}$$

where μ_0 is the prior mean and ω is the regularization parameter. $\mathbf{A}$ denotes the adjacent matrix of $\mathscr{G}$, where $\mathbf{A}_{i,j} \geqslant 0\ \forall\ i, j$. We represent the Laplacian matrix of $\mathscr{G}$ as $\mathscr{L} = \mathbf{D} - \mathbf{A}$, where $\mathbf{D} = \text{diag}(\mathbf{D})$. Then we have

$$\mathbf{r} \sim \mathscr{N}(\mu_0, \mathbf{K}_0) \tag{3}$$

where $\mu_0 = \mu_0 \times 1$ and $\mathbf{K}_0 = (\mathscr{L} + \omega I)^{-1}$. Note that this prior model is also known as Gaussian random fields.

Suppose for $S_T = \{v_1, \ldots, v_T\}$, $f(v) = r(v) + \varepsilon$, where $\varepsilon \sim \mathscr{N}(0, \sigma_n^2)$. We assume the unknown reward function $r(v)$ is sampled from a GP prior with mean 0 sand covariance function κ [57]. Therefore, for each unselected $v \in \mathbf{V}/S_t$, we can present the posterior distribution of $f(v)$ as

$$\begin{aligned} \mu_t(v) &= \mathbf{k}_t(v)^T\left(\mathbf{K}_t + \sigma_n^2 I\right)^{(-1)}\mathbf{f}_t \\ \sigma_t^2 &= \kappa(v, v) - \mathbf{k}_t(v)^T\left(\mathbf{K}_t + \sigma_n^2 I\right)\mathbf{k}_t(v) \end{aligned} \tag{4}$$

where $\mathbf{k}_t(v) = [\kappa(v, v), \ldots, \kappa(v_t, v)]^T$, $\mathbf{K}_t$ is the positive semi-definite kernel matrix where $\mathbf{K}_t(i, j) = [\kappa(v_i, v_j)]\ \forall\ i, j < t$, and $\mathbf{I}$ is the identity matrix with the size of $t \times t$. The mean and the variance will support to navigate the exploration—exploitation trade off in further steps. Finally, we leveraged the extension of the popular GP-UCB algorithm proposed by Vanchinathan et al. [36], which authors named GP-SELECT.

Algorithm 1 GP-UCB without replacement

Input: bugdet T, β_t,

1: **For** $t < T$ **do**
2: update posterior μ_{t-1}, σ_{t-1}
3: $v_t = \arg\max_{v \in \mathbf{V}/S_t} = \mu_{t-1}(v) + \beta_t \sigma_{t-1}(v)$
4: $f(v_t) = r(v_t) + \epsilon_t$
5: $S_t \leftarrow S_T \cap v_t$
6: **end For**

Output: S_T

The algorithm utilized in this work is described in Algorithm 1. The algorithm treats the exploration-exploitation dilemma with GPs. The algorithm selects the next node to query based on the selection rule at iteration t:

$$v_t = arg\,max_{v \in \mathbf{V}/S_t} = \mu_{t-1}(v) + \beta_t \sigma_{t-1}(v) \tag{5}$$

where $\mu(t)$ is the exploitation term and $\sigma(t)$ is the exploration term, and the trade-off between exploitation and exploration is handled by the time varying parameter β_t. Next, with the selected sub-dataset $S_T = \{v_1,$

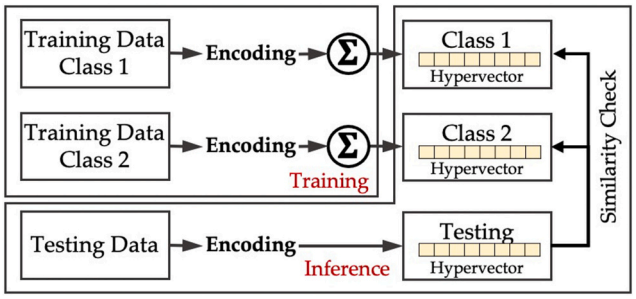

Fig. 4. Overview of hyperdimensional classification.

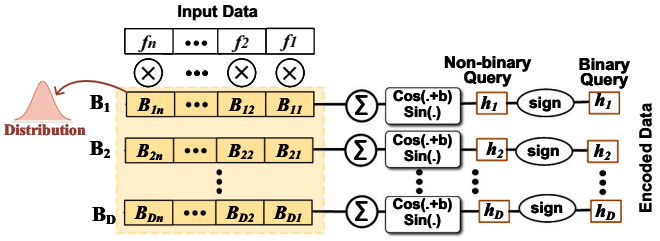

Fig. 5. Non-linear hyperdimensional encoding.

$\ldots v_T\}$, we propose a novel methodology to train a classification model with limited number of rare positives.

3.2. HDC: supervised learning using hyperdimensional computing

Fig. 4 shows the flowchart of hyperdimensional classification. HDC often contains four steps, namely encoding, single-pass training, inference, and retraining.

3.2.1. Encoding

In the first step, the algorithm maps each data points into a high-dimensional space, named encoding. HDC leverages different encoding methodologies according to the data property. HDC uses different encoding methods depending on data types [12,55,58]. The encoded data should satisfy the common-sense principle: data points different from each other in the original space should also be different in the HDC space.

In this paper, we introduce an encoding method which exploits the kernel trick [59,60] to map data points into the high-dimensional space. The underlying idea of the kernel trick is that data, which is not linearly separable in original dimensions, might be linearly separable in higher dimensions. Let us consider certain functions $K(x,y)$ which are equivalent to the dot psroduct in a different space, such that $K(x,y) = \Phi(x) \cdot \Phi(y)$, where $\Phi(\cdot)$ is often a function for high-dimensional projection. The Radial Basis Function (RBF) or Gaussian Kernel is the most popular kernel: $K(x,y) = e^{\frac{-\|x-y\|^2}{2\sigma^2}}$. We can take advantage of this implicit mapping by replacing their decision function with a weighted sum of kernels:

$$f(\cdot) = \sum_{i=0}^{N} c_i K(\cdot, x_i)$$

where (x_i, y_i) is the training data sample, and the c_is are constant weights. The study in [59] showed that the inner product can efficiently approximate RBF kernel, such that:

$$K(x,y) = \Phi(x)\cdot\Phi(y) \approx z(x)\cdot z(y)$$

The Gaussian kernel function can now be approximated by the dot product of two vectors, $z(x)$ and $z(y)$. The proposed encoding method is inspired by the RBF kernel trick method. Fig. 5 shows our encoding

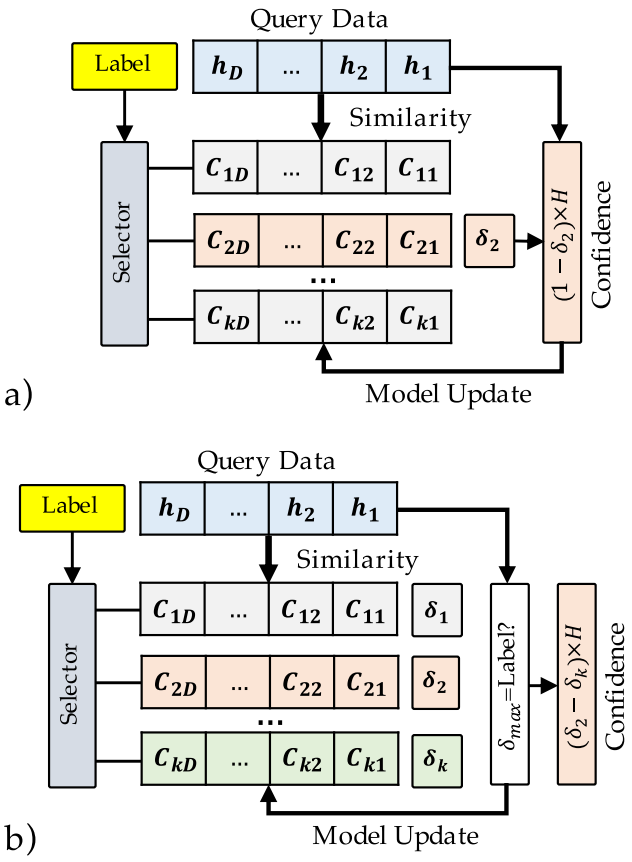

Fig. 6. (a) HDC single-pass training; and b) adaptive iterative training.

procedure. Let us consider an encoding function that maps a feature vector $\overrightarrow{\mathscr{F}} = \{f_1, f_2, \ldots, f_n\}$, with n features ($f_i \in \mathbb{R}^N$) to a hypervector $\overrightarrow{\mathscr{H}} = \{h_1, h_2, \ldots, h_D\}$ with D dimensions ($h_i \in \{-1, 1\}$). We generate each dimension of the encoded data by calculating a dot product of the feature vector with a randomly generated vector as: $h_i = cos\left(\overrightarrow{\mathscr{B}}_i \cdot \overrightarrow{\mathscr{F}} + b\right) \times sin\left(\overrightarrow{\mathscr{B}}_i \cdot \overrightarrow{F}\right)$, where B_i is the randomly generated vector with a Gaussian distribution (mean μ=0 and standard deviation σ=1) with the same dimensionality of the feature vector and b is a random value sampled uniformly from $[0, 2\pi]$.

The random vectors $\left\{\overrightarrow{\mathscr{B}}_1, \overrightarrow{\mathscr{B}}_2, \cdots, \overrightarrow{\mathscr{B}}_D\right\}$ can be generated once offline and then can be used for the rest of the classification task. After this step, each element h_i of a hypervector $\overrightarrow{\mathscr{H}}^n$ has a non-binary value. In the HDC, binary (bipolar) hypervectors are often used for the computation efficiency. We thus obtain the final encoded hypervector by binarizing it with a sign function ($\overrightarrow{\mathscr{H}} = sign\left(\overrightarrow{\mathscr{H}}^n\right)$) where the sign function assigns all positive hypervector dimensions to '1' and zero/negative dimensions to '−1'. The encoded hypervector stores the information of original data point with D bits.

3.2.2. Training

The objective of the single-pass training is to find the universal property for training dataset. The trainer model adds the hypervectors to create a single vector and linearly combines hypervectors belonging to each class. Assume we have k classes $\mathscr{M} = \left\{\overrightarrow{\mathscr{C}}_1, \overrightarrow{\mathscr{C}}_2, \ldots \overrightarrow{\mathscr{C}}_k\right\}$, if we have $\mathscr{J}$ inputs with label l, the class hypervector $\overrightarrow{\mathscr{C}}^l$ can be calculated as

$$\overrightarrow{\mathscr{C}}^{l} = \sum_{j}^{J} \overrightarrow{\mathscr{H}}^{l}_{j} \tag{6}$$

However, the above-mentioned hypervector generation often results in the saturation of class hypervectors by data points with the most common patterns and leads to misclassification. Therefore, we design a novel HDC, an adaptive training framework for efficient and accurate learning. The proposed algorithm identifies common patterns during training and eliminates the saturation of the class hypervectors during single-pass training. Instead of naively combining all encoded data, our approach adds each encoded data to class hypervectors depending on how much new information the pattern adds to class hypervectors.

Fig. 6a shows HDC functionality during adaptive initial training. Assume $\overrightarrow{\mathscr{H}}$ a new training data point. The proposed HDC first computes the cosine similarity of this new data point with all class hypervectors $\overrightarrow{\mathscr{C}}_i$ as

$$\delta\left(\overrightarrow{\mathscr{H}}, \overrightarrow{\mathscr{C}}^{l}\right) = \frac{\overrightarrow{\mathscr{H}} \cdot \overrightarrow{\mathscr{C}}^{l}}{\| \overrightarrow{\mathscr{H}} \| \cdot \| \overrightarrow{\mathscr{C}}^{l} \|} \tag{7}$$

where $\overrightarrow{\mathscr{H}} \cdot \overrightarrow{\mathscr{C}}_i$ denotes the dot product between the query and the class hypervector. HDC updates the model based on the δ similarity. For example, if the input data has a label of l but the more similar to class l' from the calculation, we update the model as follows

$$\begin{aligned} \overrightarrow{\mathscr{C}}_{l} &\leftarrow \overrightarrow{\mathscr{C}}_{l} + \eta(1-\delta_{l})\overrightarrow{\mathscr{H}} \\ \overrightarrow{\mathscr{C}}_{l'} &\leftarrow \overrightarrow{\mathscr{C}}_{l'} + \eta(1-\delta_{l'}) \times \overrightarrow{\mathscr{H}} \end{aligned} \tag{8}$$

where η represents the learning rate. When δ_l is large, i.e., the input is a common data point which is already exist in the model, the update adds a very small portion of encoded query to model to eliminate model saturation ($1 - \delta_l \simeq 0$). On the other hand, when δ_l has a small value, i.e., the query has a new pattern which does not exist in the model, we update the model with a large factor ($1 - \delta_l \simeq 1$).

3.2.3. Inference

Inference checks the similarity of each encoded test data with the class hypervector. We encode the input and generate a query hypervector $\overrightarrow{\mathscr{H}}$. The similarity δ can be calculated by $\overrightarrow{\mathscr{H}}$. Query data gets the label of the class with the highest similarity. Finally, retaining improves HDC classification accuracy by discarding the mispredicted queries from corresponding mispredicted classes and adding them to the right class.

3.2.4. Retraining

Fig. 6b) illustrates the functionality of HDC during adaptive learning. Adaptive training follows a similar learning procedure as initial training. In our proposed methodology, HDC adaptive learning is equivalent to the retraining phase, as it provides a higher chance and weight to non-common patterns to represent on the final model. Retraining examines if the model correctly returns for an encoded training data H. If the model mispredicts the label l as l', we utilize $\delta_l = \delta\left(H, \overrightarrow{\mathscr{C}}_{l}\right)$ and $\delta_{l'} = \delta\left(H, \overrightarrow{\mathscr{C}}_{l'}\right)$ to calculate the similarity of data with correct and miss-predicted classes through Eq. (9). As such, we ensure that the model is updated based on how far a training data point is miss-classified with the current model. Also, we provide separate coefficients for the true and miss-predicted labels, allowing to update each class hypervector independently.

Table 1
LPBF machine parameters setting for fabrication.

Print parameters	Value
Laser power	340 W
Scan speed	1250 mm/s
Hatching distance	0.12 mm
Layer thickness	60 μm

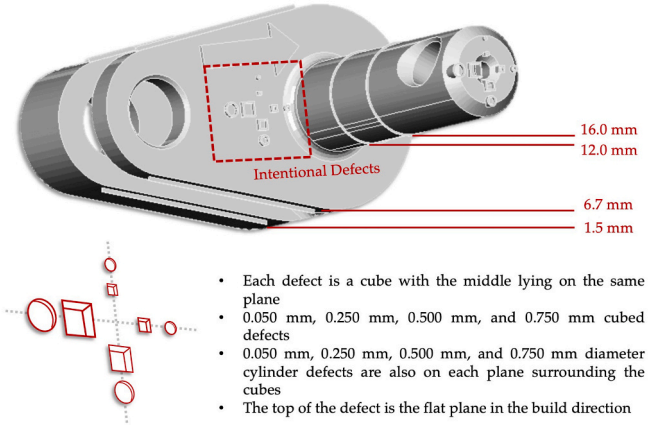

Fig. 7. : Locations of intentional defects in the drag link joint build.

4. Experimental results

4.1. Build fabrication

To evaluate and validate the proposed methodology, we fabricated a drag link joint build with the powder bed fusion (PBF) technology in which the machine utilizes a laser power source for melting the metal powder. Our experiment was performed on an EOSINT M 280 Laser PBF machine. The material was a titanium alloy, Ti-6Al-4V, also known as ASTM B348 Grade 23 powder material which has a particle size between 14 μm and 45 μm. The LPBF machine experimental setting is shown in Table 1.

Overall, the drag link joint build has an enclosing box dimension of 23.7 mm × 13.3 mm × 27.3 mm, with the layer thickness of 60 μm. We intentionally design defects in the build, and the flaws are embedded in the build at four different locations along the build-up direction that is intersected with cubical and cylindrical patterns (see Fig. 7). Flaw patterns consist of 50 μm, 250 μm, 500 μm, and 750 μm cubical and cylindrical shapes, which are centered in the z plane direction. Cylindrical flaws with the diameter of 50 μm, 250 μm, 500 μm, and 750 μm are also placed in the part. These intentionally embedded flaws represent the lack-of-fusion flaws that happen in the Laser PBF process, i.e., small zones of infused material placed in a component.

During the fabrication, a DSLR camera (i.e., Nikon D800E) with a resolution of 36.3 megapixels is utilized to capture layerwise image profiles of the powder bed (see [61] for detailed information related to the printer setup). The layerwise images captured by the DSLR camera contain both the region of interest (ROI) and the powdered area. As shown in Fig. 8c), we perform an image registration to separate the ROI from the background. Specifically, we utilized the standard registration procedure to register the moving voxel (i.e., XCT volume) and the fixed voxel (i.e., STL file) through five steps, namely interpolation, comparison, optimization, parameter update, and transformation.

First, image registration is performed with the backward mapping starts at interpolation (Fig. 9). We utilize bi-linear interpolation to map voxel values to the new coordinate system based on the moving transformation. Then, the mean square difference is defined to measure the similarity metric between fixed and transformed voxels. The gradient

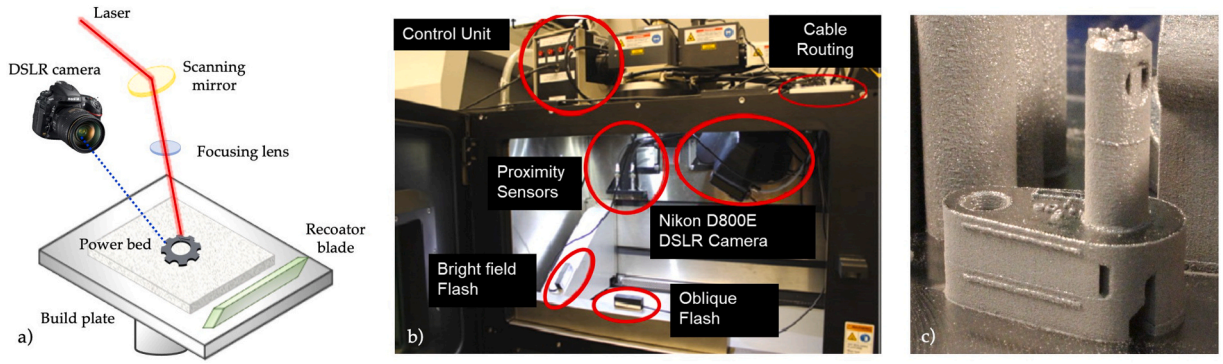

Fig. 8. : a) The schematic diagram of the LPBF process, b) the advanced imaging system, and c) a picture of the fabricated drag link joint build.

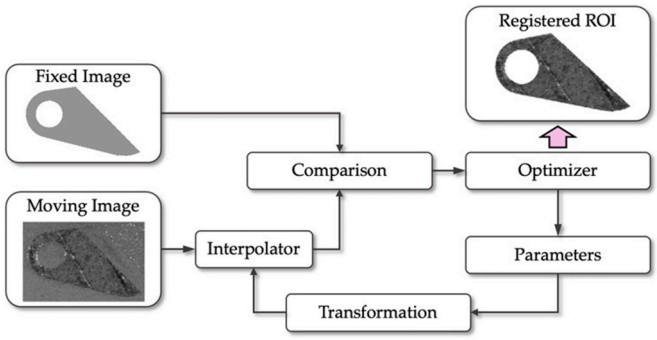

Fig. 9. : Image registration flowchart to map layerwise image according to the CAD slice.

descent optimizer specifies the searching strategy by calculating the optimal set of transformation parameters (i.e., translation, scaling, and rotation angles around z, y, and x axes). Finally, the parameters and the transformation map the position of each unit in two voxels. The algorithm is aimed at optimizing the parameter to find the spatial mapping, which brings the moving voxel to align with the fixed target.

Fig. 10 shows two examples of image registration, a) for a layer with no defect and b) for a layer with intentional defects, respectively. We can clearly see the two cubical defects are captured by the layerwise images. When the defect is first present, it is often shown as dark spots in the layerwise image. However, when fabricating the next layer, the metal powder will fall into the defect and will be presented in gray color by the layerwise image (see Fig. 1). We treat the layer with defects as rare positives and label them as 1. For more information regarding the experimental setup and iterative image registration, please refer to our previous works [62,63].

4.2. Active search for locating rare positives

We first run the active search on the selected set, and the recall (i.e., the percentage of all rare positives that are selected and annotated by bandits) is visualized in Fig. 11. Note that in total, we have 66 layers (layers 2 to 67) of the drag link joint build which contains 16 layers of rare positives and 50 negative data points for our study.

We consider the following parameter values for our experiment. $\beta = \{2,1,0.1,0.01\}$, $\omega = \{0.5,0.1,0.05,0.01\}$, $\sigma^2=\{1,0.5,0.25,0.1\}$, and $D=\{2,3,4,5\}$. Here, D denotes the maximum depth of a network structure. For example, $D=3$ indicates that a layer (i.e., vertex in $\mathcal{G}$ can be linked at most to the previous 3 and next 3 layers in the network structure. Fig. 12 shows the result of recall versus the number of queries of the whole dataset. Note that we colored the 1st, 10th, 20th, 30th, 40th, 50th, and 60th annotation result among 64 different combinations. Other results are represented in gray color. Also, some results might overlap with each other. As we can see from the result, the active search algorithm is able to find all rare positives within the first 25 iterations in most cases. Once the first rare positive is found, the other rare positives are easier to be selected by the algorithm. If one layer has defects, according to the observation, the next layer has a bigger probability of having defects. A pore often exists more than one layer; for example, the size of a pore in PBF is usually more than 100 μm, and the layer thickness that we have for our experiment is 60 μm. For the next step, we select the best parameter set for each case (i.e., different network structures) for supervised learning with the proposed HDC.

4.3. HDC for supervised learning

We separate 75% of our data for training and 25% for testing for HDC. As shown in Fig. 12, we compared the testing accuracy between different numbers of queries and the testing accuracy when training the whole dataset. The testing accuracy increases as the number of queries increases, and the accuracy slowly converge to the accuracy when training the model with the whole dataset (i.e., T=66). When T=66, the accuracy of the classification is around 0.89 no matter how d varies.

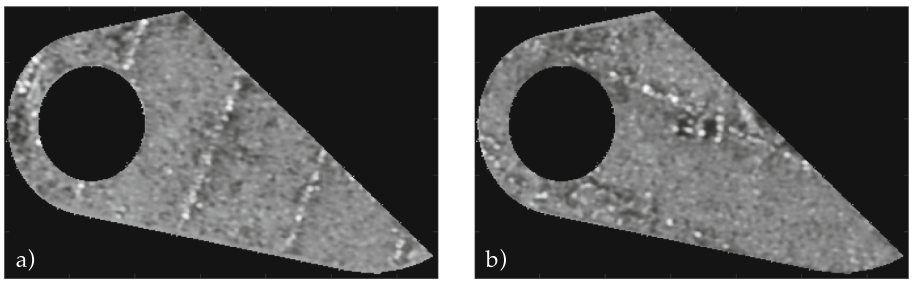

Fig. 10. Examples of registered layerwise images a) layer 5, and b) layer 51. We can see that one cubical and one cylindrical defects are captured by the layerwise image in b).

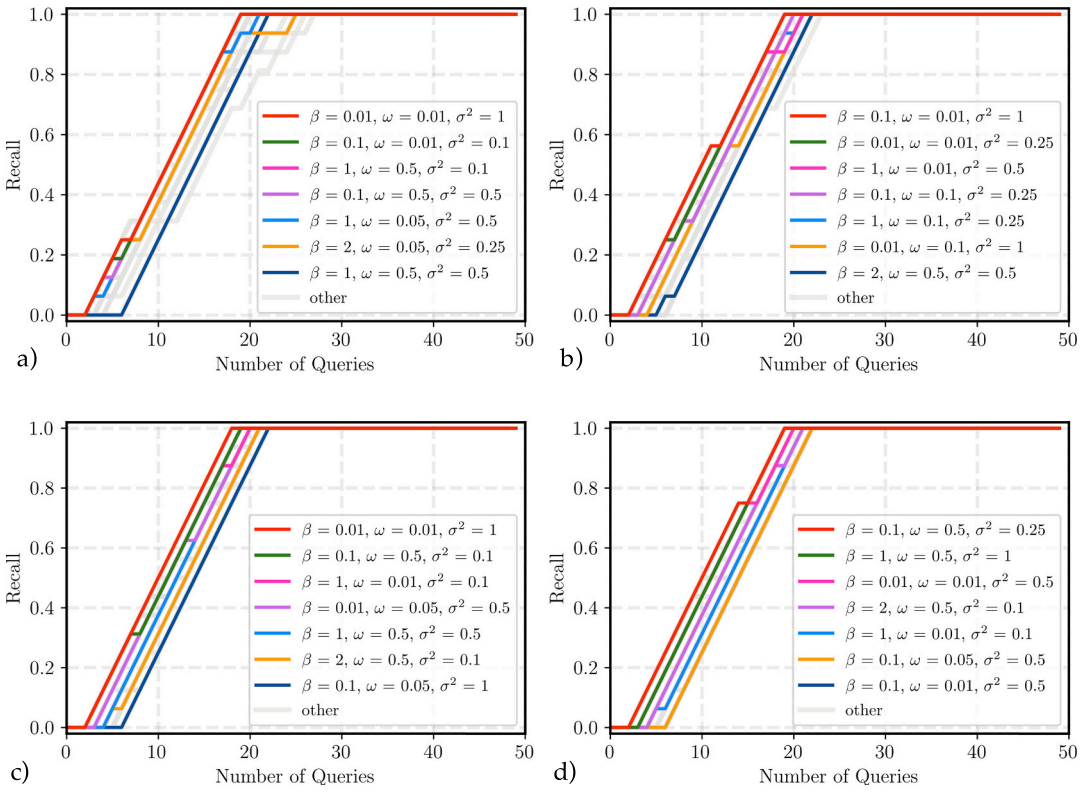

Fig. 11. The recall of the active search when a) D=2, b) D=3, c) D=4, and d) D=5.

However, the accuracy converges faster when the value of d is bigger. In our experiment, we have 16 images with the label of 1. According to Fig. 12, S_T from the active search become a balanced dataset when T>30. In addition, the variance of the accuracy decreases when the number of queries increases. This is because that the model becomes more stable when the number of training data increases. We also collect the running time of the proposed algorithm. It can be seen in Table 2 that the running time increases when the size of the dataset increases.

We further report other indicators, i.e., sensitivity (SEN), specificity (SPC), and precision (PPV), for the above four cases when T=30.

$$\begin{aligned} \mathrm{SEN} &= \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}}, \mathrm{SPE} = \frac{\mathrm{TN}}{\mathrm{FP}+\mathrm{TN}}, \\ \mathrm{NPV} &= \frac{\mathrm{TN}}{\mathrm{TN}+\mathrm{FN}}, \mathrm{PPV} = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}} \\ \mathrm{ACC} &= \frac{\mathrm{TP}+\mathrm{TN}}{\mathrm{TP}+\mathrm{TN}+\mathrm{FP}+\mathrm{FN}} \end{aligned} \tag{9}$$

where TP, FP, TN, FN denote true positive, false negative, false positive, and true negative in the confusion matrix, respectively. Here, the sensitivity indicates the ability of our model to correctly identify the rare positives, specificity shows the capability of the model to correctly identify layers without defects, and precision shows the proportion of rare positives that are identified correctly. From Table 3, the indicators do not vary much under different d. Overall, the proposed HDC enables fast and efficient learning from a small amount of data.

5. Conclusions

Additive manufacturing (AM) processes still suffer from various process-related quality issues such as cracks, porosity, geometric distortion, and delamination. The recent advancement in imaging technology leads to a data-rich environment in AM and provides a unique opportunity to enhance the understanding of processes. Machine learning (ML) shows great potential to accelerate the transition of high-dimensional image data into real-time actionable knowledge in AM. However, implementation of ML methods requires a significant amount of labeled data, which is often tedious, costly, and requires substantial human efforts with considerable expertise. This research is aimed at tangling active search strategy and neuromorphic computing to selectively label AM image data and swiftly learn their underlying dynamics. We incorporate ideas from Gaussian Process optimization and multi-armed bandits to provide a principled approach for active search with strong theoretical guarantees. Active search takes advantage of graph analysis to find rare defective layerwise images as many as possible within a given budget using exploration and exploitation formulation. Novel hyperdimensional computing is introduced to provide a natural way to scalable and strong learning system that better mimics brain functionality. The proposed active search and neuromorphic computing framework is shown to have strong potentials for analyzing in a diverse set of domains with in-situ imaging data to support learning tasks such as defect detection, process parameter optimization, and design recommendation with a lower budget. Experimental results on complex additive manufacturing build show that with 33% queries of data, the accuracy of the proposed framework converges to the accuracy when training the model with the whole dataset (i.e., $T = 66$). The proposed active search and neuromorphic computing framework is shown to have strong potentials for analyzing in a diverse set of domains with in-situ imaging data to support learning tasks such as defect detection, process parameter optimization, and design recommendation with a lower budget. Our contribution to process improvement in AM is two-fold. First, using less data through active search allows defect detection at a faster speed. Second, most machine learning methods, especially powerful deep neural networks, require a large number of training data for anomaly detection in AM. Such data are not usually available due to the change in process, design, or material types. In the future, we will further investigate and develop active search and neuromorphic methods and tools for monitoring complex spatial patterns in 3D/4D image streams.

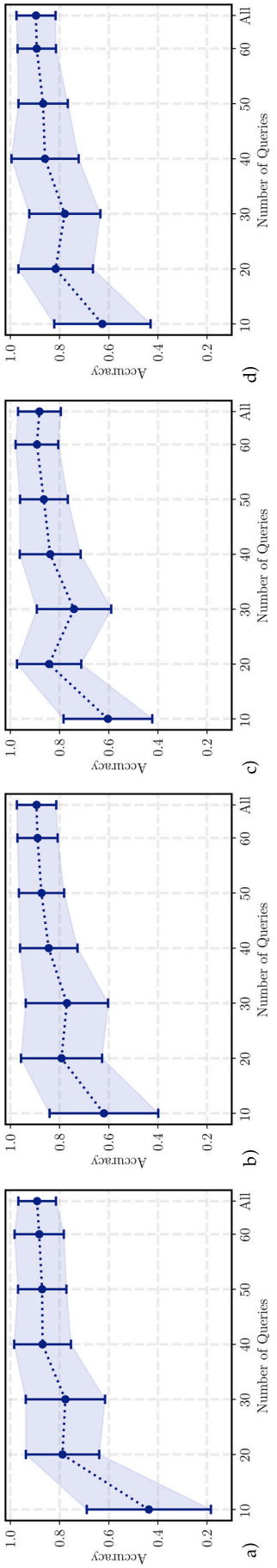

Fig. 12. Testing accuracy of the proposed model when a) d=2, b) d=3, c) d=4, and d) d=5.

Table 2
Time utilized for supervised learning by the proposed HDC under different scenarios.

Time (s)	d=2	d=3	d=4	d=5
n=10	0.416 ± 0.008	0.408 ± 0.042	0.398 ± 0.009	0.412 ± 0.009
n=20	0.929 ± 0.006	0.939 ± 0.016	0.902 ± 0.091	0.921 ± 0.008
n=30	1.375 ± 0.016	1.387 ± 0.016	1.373 ± 0.025	1.363 ± 0.008
n=40	1.880 ± 0.021	1.957 ± 0.030	1.897 ± 0.021	1.875 ± 0.015
n=50	2.423 ± 0.031	2.378 ± 0.053	2.374 ± 0.035	2.315 ± 0.023
n=60	2.852 ± 0.011	2.863 ± 0.064	2.889 ± 0.039	2.934 ± 0.059
$n = N$	3.178 ± 0.059	3.212 ± 0.050	3.296 ± 0.936	3.221 ± 0.029

Table 3
Time utilized for supervised learning by the proposed HDC under different scenarios.

Indicator	d=2	d=3	d=4	d=5
SEN	0.839	0.804	0.791	0.822
SPC	0.790	0.813	0.778	0.800
PPV	0.831	0.840	0.819	0.848

Declaration of competing interest

The authors wish to confirm that there are no known conflicts of interest associated with this publication and there has been no significant financial support for this work that could have influenced its outcome.

Acknowledgment

The authors gratefully acknowledge Dr. Edward W. Reutzel from CIMP-3D at Penn State providing the data utilized in this research. This work was partially supported by Semiconductor Research Corporation (SRC) Task No. 2988.001, Department of the Navy, Office of Naval Research, grant #N00014-21-1-2225, National Science Foundation (NSF) #2127780, and a generous gift from Cisco.

References

[1] Litjens G, Kooi T, Bejnordi BE, Setio AAA, Ciompi F, Ghafoorian M, et al. A survey on deep learning in medical image analysis. Med Image Anal 2017;42:60–88.
[2] De Bruijne M. Machine learning approaches in medical image analysis: from detection to diagnosis. Med Image Anal 2016;33:94–7.
[3] Imani F, Cheng C, Chen R, Yang H. Nested gaussian process modeling and imputation of high-dimensional incomplete data under uncertainty. IISE Trans Healthc Syst Eng 2019;9:315–26.
[4] Yurtsever E, Lambert J, Carballo A, Takeda K. A survey of autonomous driving: common practices and emerging technologies. IEEE Access 2020;8:58443–69.
[5] Grigorescu S, Trasnea B, Cocias T, Macesanu G. A survey of deep learning techniques for autonomous driving. J Field Robot 2020;37:362–86.
[6] Navarro PJ, Fernandez C, Borraz R, Alonso D. A machine learning approach to pedestrian detection for autonomous vehicles using high-definition 3D range data. Sensors 2017;17:18.
[7] Imani F. Sensor-based modeling and analysis of advanced manufacturing systems for quality improvements. 2020.
[8] Yao B, Imani F, Yang H. Markov decision process for image-guided additive manufacturing. IEEE Robot Autom Lett 2018;3:2792–8.
[9] Meng L, McWilliams B, Jarosinski W, Park H-Y, Jung Y-G, Lee J, et al. Machine learning in additive manufacturing: a review. JOM 2020;72:2363–77.
[10] How data will build the factories of the future. https://www.techerati.com/the-stack-archive/data-centre/2018/05/14/smart-manufacturing-factory-automation; 2018.
[11] M. Imani, Y. Kim, S. Riazi, J. Messerly, P. Liu, F. Koushanfar, T. Rosing, A framework for collaborative learning in secure high-dimensional space, in: 2019 IEEE 12th international conference on cloud computing (CLOUD), IEEE, pp. 435–446.
[12] A. Rahimi, P. Kanerva, J. M. Rabaey, A robust and energy-efficient classifier using brain-inspired hyperdimensional computing, in: Proceedings of the 2016 international symposium on low power electronics and design, pp. 64–69.
[13] Grasso M, Colosimo BM. Process defects and in situ monitoring methods in metal powder bed fusion: a review. Measure Sci Technol 2017;28:044005.
[14] Hann BA. Powder reuse and its effects on laser based powder fusion additive manufactured alloy 718. SAE Int J Aerosp 2016;9:209–13.

[15] Spears TG, Gold SA. In-process sensing in selective laser melting (SLM) additive manufacturing. Integr Mater Manuf Innov 2016;5:16–40.
[16] Benedetti M, Fontanari V, Bandini M, Zanini F, Carmignato S. Low-and high-cycle fatigue resistance of Ti-6Al-4V ELI additively manufactured via selective laser melting: mean stress and defect sensitivity. Int J Fatig 2018;107:96–109.
[17] Scime L, Beuth J. Using machine learning to identify in-situ melt pool signatures indicative of flaw formation in a laser powder bed fusion additive manufacturing process. Addit Manuf 2019;25:151–65.
[18] Williams CK, Rasmussen CE. Gaussian processes for machine learningvol. 2. Cambridge, MA: MIT Press; 2006.
[19] Noble WS. What is a support vector machine? Nat Biotechnol 2006;24:1565–7.
[20] Peterson LE. K-nearest neighbor. Scholarpedia 2009;4:1883.
[21] Anthony M, Bartlett PL. Neural network learning: theoretical foundations. Cambridge University Press; 2009.
[22] Tapia G, Elwany AH, Sang H. Prediction of porosity in metal-based additive manufacturing using spatial gaussian process models. Addit Manuf 2016;12:282–90.
[23] Wang C, Tan X, Liu E, Tor SB. Process parameter optimization and mechanical properties for additively manufactured stainless steel 316L parts by selective electron beam melting. Mater Des 2018;147:157–66.
[24] Scime L, Beuth J. Anomaly detection and classification in a laser powder bed additive manufacturing process using a trained computer vision algorithm. Addit Manuf 2018;19:114–26.
[25] Scime L, Beuth J. A multi-scale convolutional neural network for autonomous anomaly detection and classification in a laser powder bed fusion additive manufacturing process. Addit Manuf 2018;24:273–86.
[26] Grasso MLG, Remani A, Dickins A, Colosimo B, Leach RK. In-situ measurement and monitoring methods for metal powder bed fusion—an updated review. Measure Sci Technol 2021;32.
[27] X. Wang, R. Garnett, J. Schneider, Active search on graphs, in: Proceedings of the 19th ACM SIGKDD international conference on knowledge discovery and data mining, pp. 731–738.
[28] Pyzer-Knapp EO. Bayesian optimization for accelerated drug discovery. IBM J Res Dev 2018;62:1–2.
[29] Smailović J, Grčar M, Lavrač N, Žnidaršič M. Stream-based active learning for sentiment analysis in the financial domain. Inform Sci 2014;285:181–203.
[30] P. Nagarnaik, A. Thomas, Survey on recommendation system methods, in: 2015 2nd international conference on electronics and communication systems (ICECS), IEEE, pp. 1603–1608.
[31] Garnett R, Krishnamurthy Y, Xiong X, Schneider J, Mann R. Bayesian optimal active search and surveying. In: Proceedings of the 29th International Coference on International Conference on Machine Learning (ICML'12). Omnipress; 2012. p. 843–50. https://doi.org/10.5555/3042573.3042683.
[32] Robbins H. Some aspects of the sequential design of experiments. Bull Am Math Soc 1952;58:527–35.
[33] Auer P, Cesa-Bianchi N, Fischer P. Finite-time analysis of the multiarmed bandit problem. Mach Learn 2002;47:235–56.
[34] A. Antos, V. Grover, C. Szepesvári, Active learning in multi-armed bandits, in: International conference on algorithmic learning theory, Springer, pp. 287–302.
[35] S. Zhu, J. Coles, S. Xie, Active search using meta-bandits, in: Proceedings of the 29th ACM international conference on information & knowledge management, pp. 3493–3496.
[36] H. P. Vanchinathan, A. Marfurt, C.-A. Robelin, D. Kossmann, A. Krause, Discovering valuable items from massive data, in: Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining, pp. 1195–1204.
[37] Srinivas N, Krause A, Kakade SM, Seeger MW. Information-theoretic regret bounds for gaussian process optimization in the bandit setting. IEEE Transactions on Information Theory 2012;58:3250–65. https://ieeexplore.ieee.org/abstract/document/6138914.
[38] M. Ji, J. Han, A variance minimization criterion to active learning on graphs, in: Artificial intelligence and statistics, PMLR, pp. 556–564.
[39] Y. Ma, R. Garnett, J. G. Schneider, σ-Optimality for active learning on Gaussian random fields, in: NIPS, pp. 2751–2759.
[40] Y. Ma, T.-K. Huang, J. G. Schneider, Active search and bandits on graphs using sigma-optimality, in: UAI, p. 551.
[41] Rožanec JM, Novalija I, Zajec DP, Kenda K, Mladenić D. Knowledge modelling and active learning in manufacturing. arXiv Preprint 2021:arXiv:210702298.
[42] Zajec P, Rožanec JM, Novalija I, Fortuna B, Mladenić D, Kenda K. Towards Active Learning Based Smart Assistant for Manufacturing. In: 2021 Advances in Production Management Systems. Artificial Intelligence for Sustainable and Resilient Production Systems (APMS). Springer International Publishing; 2021. p. 295–302.
[43] Wuest T, Weimer D, Irgens C, Thoben K-D. Machine learning in manufacturing: advantages, challenges, and applications. Prod Manuf Res 2016;4:23–45.
[44] M. Imani, A. Rahimi, D. Kong, T. Rosing, J. M. Rabaey, Exploring hyperdimensional associative memory, in: 2017 IEEE international symposium on high performance computer architecture (HPCA), IEEE, pp. 445–456.
[45] H. Li, T. F. Wu, A. Rahimi, K.-S. Li, M. Rusch, C.-H. Lin, J.-L. Hsu, M. M. Sabry, S. B. Eryilmaz, J. Sohn, et al., Hyperdimensional computing with 3D VRRAM in-memory kernels: device-architecture co-design for energy-efficient, error-resilient language recognition, in: 2016 IEEE international electron devices meeting (IEDM), IEEE, pp. 1–16.
[46] Karunaratne G, Le Gallo M, Cherubini G, Benini L, Rahimi A, Sebastian A. In-memory hyperdimensional computing. In: Nature electronics; 2020. p. 1–11.
[47] M. Imani, P. Saikishan, G. Saransh, Z. Minxuan, K. Yeseong, R. Tajana, Dual: Acceleration of clustering algorithms using digital-based processing in-memory, in: Proceedings of the 2020 53rd annual IEEE/ACM international symposium on microarchitecture (MICRO), vol. 1, IEEE Computer Society, pp. 356–371.
[48] Kanerva P. Hyperdimensional computing: an introduction to computing in distributed representation with high-dimensional random vectors. Cogn Comput 2009;1:139–59.
[49] D. Kleyko, E. Osipov, Brain-like classifier of temporal patterns, in: 2014 International Conference on Computer and Information Sciences (ICCOINS), IEEE, pp. 1–6.
[50] Gallant SI, Okaywe TW. Representing objects, relations, and sequences. Neural Comput 2013;25:2038–78.
[51] Karunaratne G, Schmuck M, Le Gallo M, Cherubini G, Benini L, Sebastian A, et al. Robust high-dimensional memory-augmented neural networks. Nat Commun 2021;12:1–12.
[52] Karunaratne G, Gallo ML, Cherubini G, Benini L, Rahimi A, Sebastian A. In-memory hyperdimensional computing. arXiv Preprint 2020;3:327–37.
[53] Moin A, Zhou A, Rahimi A, Menon A, Benatti S, Alexandrov G, et al. A wearable biosensing system with in-sensor adaptive machine learning for hand gesture recognition. Nat Electron 2021;4:54–63.
[54] B. Emruli, R. W. Gayler, F. Sandin, Analogical mapping and inference with binary spatter codes and sparse distributed memory, in: The 2013 international joint conference on neural networks (IJCNN), IEEE, pp. 1–8.
[55] A. Rahimi, S. Benatti, P. Kanerva, L. Benini, J. M. Rabaey, Hyperdimensional biosignal processing: a case study for emg-based hand gesture recognition, in: 2016 IEEE international conference on rebooting computing (ICRC), IEEE, pp. 1–8.
[56] Rasanen O, Saarinen J. Sequence prediction with sparse distributed hyperdimensional coding applied to the analysis of mobile phone use patterns. In: IEEE transactions on neural networks and learning systems; 2015. p. 1–12.
[57] C. E. Rasmussen, Gaussian processes in machine learning, in: Summer school on machine learning, Springer, pp. 63–71.
[58] M. Imani, J. Messerly, F. Wu, W. Pi, T. Rosing, A binary learning framework for hyperdimensional computing, in: 2019 Design, automation & test in Europe conference & exhibition (DATE), IEEE, pp. 126–131.
[59] A. Rahimi, B. Recht, Random features for large-scale kernel machines, in: Advances in neural information processing systems, pp. 1177–1184.
[60] B. Schölkopf, The kernel trick for distances, in: Advances in neural information processing systems, pp. 301–307.
[61] Imani F, Chen R, Diewald E, Reutzel E, Yang H. Deep learning of variant geometry in layerwise imaging profiles for additive manufacturing quality control. J Manuf Sci Eng 2019;141:111001.
[62] F. Imani, R. Chen, E. Diewald, E. Reutzel, H. Yang, Image-guided variant geometry analysis of layerwise build quality in additive manufacturing, in: International manufacturing science and engineering conference, vol. 58745, American Society of Mechanical Engineers, p. V001T02A041.
[63] Chen R, Imani F, Reutzel E, Yang H. From design complexity to build quality in additive manufacturing—a sensor-based perspective. IEEE Sens Lett 2018;3:1–4.