A data-driven approach for predicting printability in metal additive manufacturing processes

William Mycroft[1] · Mordechai Katzman[1] · Samuel Tammas-Williams[2,4] · Everth Hernandez-Nava[2] ·
George Panoutsos[3] · Iain Todd[2] · Visakan Kadirkamanathan[3]

Abstract

Metal powder-bed fusion additive manufacturing technologies offer numerous benefits to the manufacturing industry. However, the current approach to printability analysis, determining which components are likely to build unsuccessfully, prior to manufacture, is based on ad-hoc rules and engineering experience. Consequently, to allow full exploitation of the benefits of additive manufacturing, there is a demand for a fully systematic approach to the problem. In this paper we focus on the impact of geometry in printability analysis. For the first time, we detail a machine learning framework for determining the geometric limits of printability in additive manufacturing processes. This framework consists of three main components. First, we detail how to construct strenuous test artefacts capable of pushing an additive manufacturing process to its limits. Secondly, we explain how to measure the printability of an additively manufactured test artefact. Finally, we construct a predictive model capable of estimating the printability of a given artefact before it is additively manufactured. We test all steps of our framework, and show that our predictive model approaches an estimate of the maximum performance obtainable due to inherent stochasticity in the underlying additive manufacturing process.

Keywords Additive manufacturing · Machine learning · Powder bed fusion · Electron beam melting · Printability analysis

Introduction

Additive manufacturing technologies are critical drivers for innovation and offer potential business benefits to the industrial sector (Azam et al. 2018; Mueller 2012; Santos et al. 2006). High value manufacturing companies aim to produce additively manufactured parts that form critical components for numerous industries. Not only are such technologies ideal for rapid prototyping and building bespoke components, but

they also allow for the formation of geometries that may be difficult or impossible to construct via more conventional technologies. Consequently, additive manufacturing technologies can be used to produce parts capable of greater performance.

One of the key challenges to make this technology robust and cost-effective is to ensure that the designed components are additively manufacturable. This is typically decomposed into two properties, *printability* and *fragility*. The *printability* of an object is the ability of a given additive manufacturing process to produce a faithful realization of the object. The *fragility* of the manufactured part is the ability of the object to withstand post-printing processing and normal usage.

At the present time, *metal powder bed fusion* technology is state-of-the-art for additively manufacturing high-performance components (King et al. 2015). Such machines operate by repeatedly spreading a thin layer of powdered material across the build plate and melting this material using a heat source. In this study we focus on electron beam melting (EBM), a process that can build parts at a relatively high speed due to electromagnetic-driven mechanisms directing the beam and is capable of maintaining simultaneous melt

✉ William Mycroft
    william.mycroft@gmail.com

[1] School of Mathematics and Statistics, University of Sheffield, Sheffield, UK

[2] Department of Materials Science and Engineering, University of Sheffield, Sheffield, UK

[3] Department of Automatic Control and Systems Engineering, University of Sheffield, Sheffield, UK

[4] Department of Maritime and Mechanical Engineering, Liverpool John Moores University, Liverpool, UK

spots on the powder bed while preheating the entire layer if needed. This extra heat applied can in turn contribute to anneal parts, minimising residual stresses. Once the build process is complete, unsintered powder is removed from built surfaces and recovered for further sieving and reuse. The resulting part can then be checked for printability and fragility.

The printability defects encountered when using this technology can be classified into two distinct groups. The first of these consists of the large-scale deformations such as warping and twisting due to the stresses on the part during the build process (Denlinger and Michaleris 2016; Li et al. 2018). Such deformations frequently arise when building large flat regions, or regions where the component thickness changes rapidly. The second group contains the small-scale defects, where small regions are visibly damaged, deformed, or fail to print all together.

The large-scale defects are reasonably well understood and can be predicted using physics based modelling (Schoinochoritis et al. 2015). In this paper we focus primarily on the small-scale defects. It is well known that many geometric features tend to impact the small-scale printability and fragility of the build part. Problematic geometric features include overhanging regions, small holes, thin walls and wires, sharp edges and curved features.

To add to the complexity, the problems of printability and fragility are not separable at the level of the geometry: while geometric features exhibiting one of these properties in isolation may build without defects, a geometric feature involving multiple of these properties may not. For example, inclined planes typically need a larger thickness as the inclination increases.

Due to the underlying physics involved in the build process, printability and fragility properties are extremely stochastic. As an example, the exact sizes, shapes and orientations of the scattered material particles can affect the build process in non-trivial ways. This complexity renders physics-based simulations infeasible.

At present, manufacturing companies rely on engineering know-how and ad-hoc rules (Jee and Witherell 2017; Mani et al. 2017) to determine which geometric structures are additively manufacturable and to set operating conditions for the manufacturing processes. Moreover, due to the time and expense needed for such processes, these rules often err on the side of caution, necessarily limiting performance of the component. There is thus a need for a systematic approach to predicting performance of additive manufacturing processes.

Over the last decade, a vast body of literature has emerged of contexts where *machine learning algorithms*, algorithms which learn from data, have been able to achieve or exceed human-level performance (Libbrecht and Noble 2015; Pham and Afify 2005; Monostori et al. 1996). Modern machine learning algorithms are able to learn highly complex non-linear relationships between predictor and target variables, even in highly stochastic environments.

Recent applications of machine learning to additive manufacturing (Aminzadeh and Kurfess 2018; Kwon et al. 2018) have focused on quality detection during or post-manufacturing or on optimisation problems around build process parameters (Panda et al. 2016; Yicha et al. 2015). To the best of our knowledge, a fully systematic approach to printability analysis, allowing identification of problematic regions *before* manufacture does not exist.

In this paper we detail an end-to-end data-driven framework for determining the geometric limits of printability in additive manufacturing processes. Firstly, we detail a methodology for constructing test artefacts exhibiting a wide range of geometric features: some straightforward and others beyond the horizon of printability. Secondly, we devise a metric to assess how faithfully a component has built within a region of interest. We can use this metric to label parts as having built successfully or unsuccessfully. Finally, we use a collection of handcrafted geometric features and suitable machine learning techniques to estimate the printability of our test artefact. A schematic of our framework is shown in Fig. 1. We show that the performance of our predictive model approaches an estimate of the obtainable limit due to inherent stochasticity of the underlying additive manufacturing process in both a cross-validated sense and on a disjoint hold-out set. Since all components are from the same build, it is possible that slight bias to build idiosyncrasies has occurred, although we expect these to be small. In future work, we intend to validate these results on a wide range of test artefacts, including those from separate builds, of varying complexities and thus theoretical performance limits.

Problem formulation

As discussed, printability is a complicated property which depends on a multitude of factors including the substance and quality of powdered material, the additive manufacturing technology, the build process parameters and the geometry. In this paper, we fix all but one of these factors and focus on the impact of the geometry on the surface printability.

To make our problem tractable, we make the following assumptions about the small-scale printability. Given the physics underlying the build process, these assumptions seem reasonable.

1. Printability is a property of a small neighbourhood around the region in question. In other words, geometries far from our region of interest do not affect its printability.
2. Printability is a property of the region's surface. Thus, any defects will be visible by studying the surface of the built component.
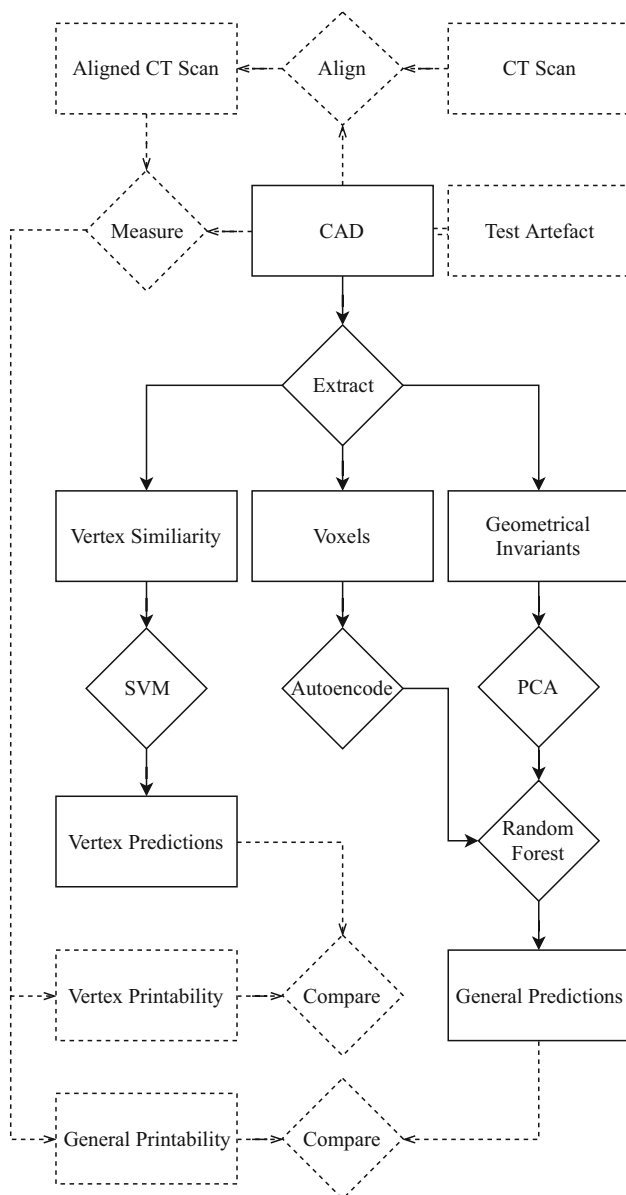
Fig. 1 Workflows for our algorithm for training (dotted and solid lines) and classification (solid lines). Rectangles denote variables and diamonds denote processing units

The inherent stochasticity and multitude of factors involved in the manufacturing process renders a physics-based modelling approach infeasible, and so we shall apply a data-driven strategy to understand the causes and also predict small-scale defects arising in additive manufacturing processes.

This strategy combined with our printability assumptions swiftly leads to a supervised learning model which takes numerical descriptors of the geometry of small regions of our *computer-aided design* (CAD) as input and outputs a quantitative printability measure for that region.

Methodology

A supervised machine learning algorithm requires data from which it is able to learn to impact of the predictor variables on the target variables. First, we must construct the parts we will use to generate our data. This process involves building and scanning bespoke custom components which are designed to contain a broad range of *local geometries*, the geometry in a small neighbourhood around a point of interest, including those sufficiently intricate to infer the limits of the additive manufacturing process.

Once we have designed our custom components, we extract invariants which describe our geometry in regions of interest and will form the input to our machine learning algorithm. To assess the printability of a given geometry, we measure the difference between our CAD and a *computed tomography* (CT) scan of the manufactured object. This quantity is the target output of our algorithm.

Armed with these geometric descriptors and measures of printability, we look to construct a predictive model. We discuss which machine learning algorithms will be most applicable to our problem and determine appropriate measures to evaluate the predictive performance.

Data generation and processing

In order to study the printability of various geometries, we must build the geometries and then obtain a detailed surface scan in order to compare them to our ground truth, the corresponding CAD. Since the additive manufacturing and scanning processes are both expensive and time-consuming, we seek to do this efficiently, gaining maximal information about the printability of various geometries for a small number of builds and scans. In this section, we detail the methods we use to generate our data.

Test artefacts To determine the capabilities of an additive manufacturing process, it is common to build a standard part which exhibits various of the aforementioned problematic features (Moylan et al. 2012). Such a part is commonly referred to as a *test artefact* and many examples exist in the literature (Mahesh et al. 2004; Kruth et al. 2005; Delgado Sanglas 2009; Moylan et al. 2014).

However, all test artefacts in the literature exhibit only a small number of geometric features in isolation. This is partly intentional; the test artefacts have been designed to ensure that standard parts will build successfully, rather than push the process to its geometric limits. While it is impossible to design an artefact containing all possible local geometries, we can hope to construct an artefact which covers this space in the sense that any possible local geometry is in some sense 'nearby' a local geometry exhibited in the artefact.

Computer designed components for additive manufacturing processes are typically specified as polyhedra: a solid

with flat polygonal faces and straight edges. For inspiration in how to construct a complex test artefact, we recall that any polyhedron can be decomposed as the union of convex polyhedra (Szilvási-Nagy 1986). The theory of random convex polyhedra is well established (Schneider 2008) and by taking unions of such objects we can generate random (not necessarily convex) polyhedra.

One model (Schneider 2008) of a random convex polyhedron is the convex hull of n points sampled from a ball of radius r. Typically, we draw n from a Poisson distribution and r from an exponential distribution. To construct a random polyhedron T, we can generate N such convex polyhedrons, apply random translations and then take the union.

The inherent stochasticity in this process produces extremely intricate geometries. Moreover, since any polyhedron can be decomposed as a union of convex polyhedra (Chazelle 1981) this process covers the space of local geometries in the sense that the support of the probability distribution contains all local geometries. To add reasonable constraints, such as ignoring parts with close to zero thickness which will definitely not print, we can exclude the convex polyhedra not meeting appropriate properties from the union.

Alignment The problem of aligning CT scan data with our CAD is non-trivial. The conventional method used for such an alignment is the *iterative closest point* (ICP) algorithm (Besl and McKay 1992), with initial transformation given by applying *Procrustes analysis* (Gower 1975) to a set of hand-picked landmark points.

In additive manufacturing processes, global deformation can be fairly common, especially when components are not suitably stress relieved after manufacture. While this effect is fairly well understood and predictable via finite element modelling, it makes the alignment process more complex as it can preclude the existence of a global alignment transformation.

As we are interested only in local measures of printability, we wish to find the optimal alignment of our CAD with our CT scan in a neighbourhood of our point of interest. To this end, we propose the following procedure. First, obtain an approximate global transformation by Procrustes analysis and the ICP algorithm. Then for every point of interest, extract the surface from the CT scan which falls within some distance r and then use the ICP algorithm to optimally align this local region with our CAD.

Finding the optimal choice of distance parameter r is the usual bias-variance trade-off: if r is too small then every region can be aligned perfectly, even in the presence of defects and if r is too large then the global deformations preclude being able to find the optimal local alignment.

Another advantage of this local alignment method is it permits having variable scaling factors across different regions of the object. Due to inaccuracies of the CT scanning process, the measurement scale is only approximately constant across the range of the scan and using a variable scaling factor allows us to correct for this. The best method for defining the uncertainty in CT spatial measurements is still debated, but inaccuracies can arise from a range of sources. These include beam hardening, cone beam effects and inaccuracies in stage movement.

Geometric descriptors

Our desired input to our machine learning algorithm is a description of the local geometry around our point of interest and in this section we define some appropriate invariants. The industry standard for a CAD in additive manufacturing processes is a *stereolithography* file. This specifies a finite triangulated mesh $\partial T \subseteq \mathbb{R}^3$: a collection of vertices V, edges E and triangular faces F. This triangulation is required to satisfy two conditions. Firstly, it is *manifold* in that there is no one-dimensional boundary. Secondly, our triangulation must be *orientated*, i.e. have a well-defined interior and exterior. We shall write T for the union of ∂T together with this interior.

The first geometric property we discuss is a measure encoding the similarity of vertices. We then discuss geometric invariants defined at the point of interest and finally, voxelisations around the point of interest.

Intuitively, the printability of a region should be invariant under rotations around the z-axis and reflections in a plane containing the z-axis as gravity is the only force acting during the build process. Thus, in order to impose this constraint on our machine learning algorithm, it is desirable that our descriptors are invariant under the action of the orthogonal group $O(2)$, acting on $\mathbb{R}^3$ by fixing the z-axis.

Vertex similarity A *spherical polygon* is a closed, connected geometric figure on the surface of the unit sphere $S^2 = \{x \in \mathbb{R}^3 \mid ||x||_2 = 1\}$ (where $||(x_1, x_2, x_3)||_2 = (x_1^2 + x_2^2 + x_3^2)^{1/2}$ denotes the Euclidean norm) whose boundary is formed by finitely many arcs of great circles. we denote the set of spherical polygons by $P(S^2)$ and the area of $P \in P(S^2)$ by $|P|$.

The infinitesimal geometry of a given vertex $v \in V$ is uniquely determined by a spherical polygon P_v as follows. Since V is a finite set, we can find a spherical neighbourhood, $B_\varepsilon(v) = \{x \in \mathbb{R}^3 \mid ||x - v||_2 < \varepsilon\}$ such that all faces which intersect this neighbourhood contain v as a vertex. The intersection $\partial T \cap \partial B_\varepsilon(v)$, projected onto the unit sphere, forms a spherical polygon which we denote P_v (Fig. 2).

Recall the *Lebesgue space* $L^2(S^2)$, a Hilbert space consisting of square integral functions on the sphere. We have a canonical embedding $P(S^2) \hookrightarrow L^2(S^2)$, mapping P to the indicator function on P given by $\chi_P(x) = 1$ if $x \in P$ and 0 otherwise. The induced inner product on $P(S^2)$ is given by $\langle P, P' \rangle = |P \cap P'|$. Unfortunately for our purposes, we do not have $\langle \tau(P), P' \rangle = \langle P, P' \rangle$ for $\tau \in O(2)$; this
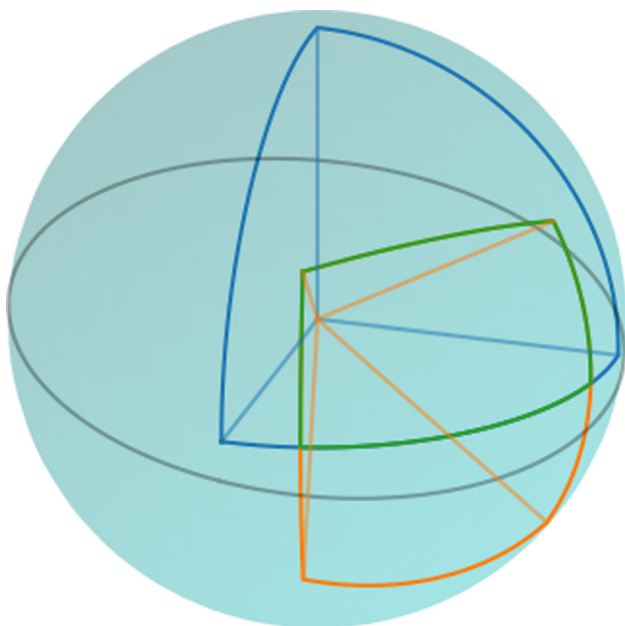
Fig. 2 Two spherical polygons (blue, orange) formed by vertices where three and four faces meet respectively. Also shown is the intersection (green), defining the vertex similarity (Color figure online)

inner product is not invariant under the action of $O(2)$ of $P(S^2)$. However, using this inner product as motivation we can define a normalised, $O(2)$-invariant similarity measure on our set of vertices.

We define the *similarity* of vertices $u, v \in V$ by

$$\kappa(u, v) = \sup_{\tau \in O(2)} \frac{|\tau(P_u) \cap P_v|^2}{|P_u||P_v|}. \tag{1}$$

Curvature The theory of estimating the curvature of a triangulated mesh has a vast quantity of literature (Panozzo et al. 2010; Chen and Schmitt 1992; Rusinkiewicz 2004). We choose to work with the definitions in Cohen-Steiner and Morvan (2003) since the constructions are easy to compute, intrinsic to properties of the triangulated mesh and apply immediately to any point in ∂T.

The *vertex defect* at a vertex $v \in V$, denoted $g(v)$, is defined to be 2π minus the sum of the interior angles (at v) of the faces which contain v. The *signed angle* at an edge $e \in E$, written $\beta(e)$, is the angle between the two faces which share the edge e taken to be positive if this angle is convex (w.r.t. the orientation) and negative otherwise. Denote the length of a line segment s by $l(s)$. For a radius r, the *Gaussian curvature measure* at $x \in \partial T$ is given by

$$\phi_r^G(x) = \sum_{v \in V \cap B_r(x)} g(v) \tag{2}$$
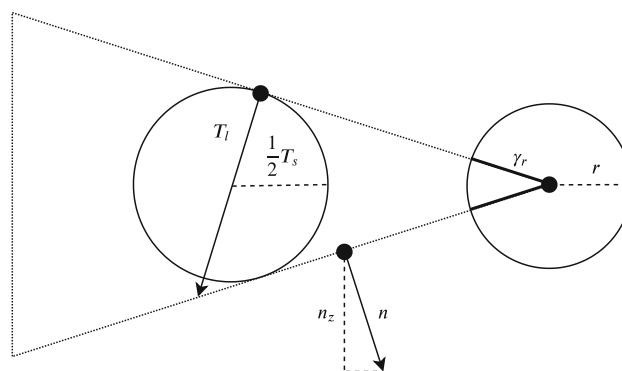
and the *mean curvature measure* by

Fig. 3 Constructions of our ray thickness T_l, sphere thickness T_s, overhang n_z, and complexity γ_r

$$\phi_r^M(x) = \sum_{e \in E} \beta(e) l(e \cap B_r(x)). \tag{3}$$

Thickness and reach The question of what is meant by the thickness of a triangulated mesh can be interpreted in many different ways resulting in many constructions and different properties (Patil and Ravi 2006; Yezzi and Prince 2003). We focus on the methods of the maximal inscribed sphere, and the maximal length of a ray cast into the object. In Inui et al. (2016), an efficient algorithm from computing the maximal inscribed sphere is provided.

The *sphere thickness* of ∂T at x is the diameter of the maximal sphere inscribed in T which is tangent to ∂T at x. The *ray thickness* is the maximal length of a ray cast from x in the opposite direction of the outward facing normal at x which does not intersect ∂T. We define the *sphere reach* and the *ray reach* of ∂T at x to be the thickness of $\partial(\mathbb{R}^3 - T)$ at x (Fig. 3).

Overhang and complexity The overhang of a region is known to correlate highly with printability (Mani et al. 2017). This is encoded by the average z-component of the normal to the surface. Additionally, the amount of surface area of the mesh nearby a point encodes the local complexity of a region. Explicitly, letting $n_z(y)$ denote the z-component of the surface normal at $y \in \partial T$, we define the *r-overhang* of ∂T at x by

$$\eta_r^z(x) = \frac{1}{|\partial T \cap B_r(x)|} \int_{\partial T \cap B_r(x)} n_z(y) dy \tag{4}$$

and the *r-complexity* by

$$\gamma_r(x) = \frac{|\partial T \cap B_r(x)|}{\pi r^2} \tag{5}$$

where $|X|$ denotes the area of X. Since our surfaces are triangulated meshes, it is computationally efficient to exactly compute these properties.
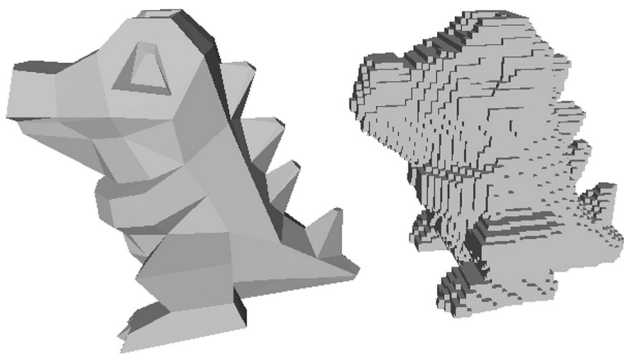
Fig. 4 An example mesh and realisation of a voxel map produced by the subdivision algorithm

Voxels While we expect the geometric features discussed to correlate with printability, they are far from producing a complete descriptor of the local geometry at non-vertex points. While triangulated meshes are a popular combinatorial data structure for describing a three-dimensional object, there is no method to describe the neighbourhood of a point as a fixed length vector. An alternative common data structure to describe three-dimensional objects is via voxel maps, three-dimensional analogues of pixels (Fig. 4).

A *voxel map* is a subset $V \subseteq \mathbb{Z}^3$, together with an *origin* $o \in \mathbb{R}^3$, and a *pitch* $\varepsilon \in \mathbb{R}$. The *realisation* of a voxel map $\mathcal{R}(V, o, \varepsilon) \subseteq \mathbb{R}^3$ is the union of the set of cubes enumerated by $v \in V$ with side length ε and centre $v + o$. Explicitly, we define $\mathcal{R}(V, o, \varepsilon)$ to be the set

$$\{x \in \mathbb{R}^3 \mid \exists v \in V \text{ s.t. } ||x - o + \frac{v}{\varepsilon}||_\infty \leq \varepsilon/2\} \quad (6)$$

where $|-|_\infty$ denotes the *supremum norm*: $||(x_1, x_2, x_3)||_\infty = \max(|x_1|, |x_2|, |x_3|)$.

There are numerous algorithms for producing voxel maps from triangulated meshes (Jones 1996; Huang et al. 1998). A simple, efficient algorithm which preserves connectivity is the subdivision algorithm: repeatedly subdivide the mesh until every edge is shorter than half the pitch and then snap each of the vertices of the resultant mesh to the nearest point on the 3D lattice $o + \varepsilon\mathbb{Z}^3$. A simple adaptation of this algorithm allows us to give a description of the geometry in the neighbourhood of a point of interest.

Given a point $x \in T$, a *radius* r and a *pitch* ε we define the *voxel neighbourhood* $\mathcal{N}(x; r, \varepsilon) \in \{0, 1\}^{\times(2r+1)^3} \subseteq \mathbb{R}^{(2r+1)^3}$ by applying the subdivision algorithm with origin x and pitch ε to voxelise ∂T. Denoting the resultant voxel map by V, we define $\mathcal{N}(x; r, \varepsilon)$ by

$$\mathcal{N}(x; r, \varepsilon)_{i,j,k} = \begin{cases} 1 & \text{if } (i, j, k) \in V \\ 0 & \text{if } (i, j, k) \notin V \end{cases} \quad (7)$$

where $|i|, |j|, |k| \leq r$. We define $r\varepsilon$ to be the *size* of the neighbourhood.

For a fixed neighbourhood size, these neighbourhoods become more complete descriptors of the local geometry as ε tends to zero. Unfortunately, this causes r, and thus the dimension of our feature space to tend to infinity. Consequentially, finding our optimal values of r and ε for our learning task will be the usual bias-variance trade-off encountered in supervised learning problems.

These voxel neighbourhoods are not invariant under our action of $O(2)$ and there is no natural action of $O(2)$ on the space of voxel neighbourhoods. However, the space of voxel neighbourhoods admits an action of the group of symmetries of the square $D_4 \subseteq O(2)$ in the obvious fashion. This will allow us to encode invariance under the action of D_4 via the technique of *data augmentation*.

Printability measure

To assess the printability of a part, we need a numerical quantity which encodes the difference between our CAD ∂T and the corresponding build P. We assume that the CT scan produces a faithful representation of the built component and that we have already locally aligned our scan with our CAD as described in "Data generation and processing" section.

For each point $x \in \partial T$, we define the *Hausdorff printability measure*, illustrated in Fig. 5, by

$$\rho(x) = \inf_{y \in P} ||y - x||_2 \in [0, \infty). \quad (8)$$

For a fixed threshold t, we define a point as *problematic* if $\rho(x) > t$ and *printable* if $\rho(x) \leq t$.

The main advantages of this printability measure are its simplicity, computational ease and interpretability. The obvious drawback is that it is a point-wise measure of printability; while we might have $\rho(x) = 0$, we may still have a printing defect in the region of x. However, when sampling points at random from M this will in general provide an accurate measure of the (lack of) printability at the given point.

We choose to focus on the classification problem of predicting whether a point will be printable or problematic, rather than the regression problem of estimating the exact value of $\rho(x)$. Our motivation for this is twofold: it allows us to distinguish genuine build defects from the natural surface roughness produced by additive manufacturing processes and moreover there are often industrial requirements; for many applications it is simply desired that components comply with a given specification. Our printability threshold t can be adjusted by the end-user as appropriate for the given application, for example to meet functional constraints.
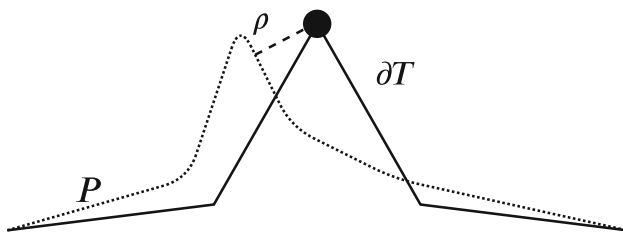
Fig. 5 Illustration of our printability measure ρ at a point, the CAD model and build are represented by solid and dotted lines respectively

Machine learning techniques

There is an abundance of literature on machine learning algorithms and techniques for applying them. In this section, we discuss which algorithms are most suited to handle our geometric invariants and provide references for the precise inner workings of such algorithms.

Support vector machines Certain machine learning algorithms naturally take as input a similarity measure, such as our vertex similarity measure. SVMs (Bishop 2013) are a computationally efficient such algorithm which can learn complex classification rules via the *kernel trick*, implicitly mapping the inputs into a high, or infinite, dimensional space where distances correspondence to a precomputed similarity measure (Chen et al. 2009). A major advantage of support vector machines is their *sparsity*, to make predictions we only need to compute our similarity measure with a small subset of the training data.

Random forests Random forests (Hastie 2009) are built as an ensemble of extremely simple rule based classifiers, known as decision trees, which require limited data pre-processing and are easily interpretable. However, to learn complex structure decision trees must be grown very deep and often have poor generalisation performance. A random forest combats this by building many decision trees on random subsets of the data and making classifications based on majority vote.

Principal component analysis Many classification algorithms are subject to the *curse of dimensionality*. Moreover, many of our geometric invariants are highly correlated and so the dimensionality of our feature space is high, even if the data tends to lie in a lower dimensional subspace. To this end, we employ *principal component analysis* (PCA) (Bishop 2013), a linear technique which transforms observations of possibly correlated variables into uncorrelated variables known as *principal components*. The transformation is defined so that the first principal component has the largest possible variance, and each following component has the highest possible variance subject to being orthogonal to the proceeding components. This is used as a dimensionality reduction technique by retaining only the first N principal components.

Autoencoders Our voxel invariants, while being close to complete descriptors of the local geometry, are inherently high-dimensional. Moreover, important geometric properties are likely to be non-linear. A powerful non-linear dimensionality reduction technique is the *autoencoder* (Goodfellow 2016). Autoencoders are neural networks for which the target output is the same as the input. Their architecture forces them to compress the input into a compact latent representation, before reconstructing the output with minimal information loss. Convolutional layers can be used with the neural network to encode translational invariance of features. These have been shown to produce exceptional performance on 2 and 3 dimensional image data, i.e. voxel maps.

Data augmentation An additional technique which shall use for two different purposes is data augmentation. Firstly, as we shall discuss, out of the box supervised learning algorithms often struggle with imbalanced datasets. A simple way to combat this is to oversample or overweight under-represented classes to augment the data set. A more sophisticated method of oversampling we employ is the *synthetic minority over-sampling technique* (SMOTE) (Bowyer et al. 2011). Another application of data augmentation is the problem of learning invariance under group actions (Gao and Ji 2017), such as our action of D_4 on our voxel descriptors. Augmenting the predictor variables with their images under the actions of the group assists an algorithm in learning this invariance.

Model selection

To study which particular predictive models and corresponding hyperparameters perform best on our classification task, we need a metric by which we can assess the resulting performance. For reasons we shall discuss, traditional metrics are not well-suited to our problem and so we generalise these to formulations more appropriate to our context.

The problem of labelling areas of our CAD as printable or problematic is an example of an *imbalanced learning problem* (He and Garcia 2009): the number of printable regions will in general drastically outweigh the number of problematic regions. The conventional measure of performance for classification problems is accuracy, the percentage of examples that are classified correctly. In the imbalanced situation, this metric is ill-suited: an algorithm that simply predicts the modal class for every example will achieve a high accuracy, but is useless for prediction. Metrics derived from the *confusion matrix* are much more appropriate in this setting.

Suppose our confusion matrix has true and false positives and true and false negatives denoted by **tp**, **fp**, **tn** and **fn** respectively. The *recall* is defined by $\mathbf{tp}/(\mathbf{tp} + \mathbf{fp})$ and the *precision* by $\mathbf{tp}/(\mathbf{tp} + \mathbf{fn})$. The standard way of combining these two metrics is by the F_β-measure, given by

$$F_\beta = (1 + \beta^2) \cdot \frac{\text{precision} \cdot \text{recall}}{(\beta^2 \cdot \text{precision}) + \text{recall}}. \tag{9}$$

The parameter β controls the relative importance of recall and precision. The choice of value of β is very important from a business perspective; increasing β will increase the likelihood of correctly capturing all problematic regions, at the additional time expense of having to handle additional false positives.

By construction, the F_β-measures are well-suited to imbalanced learning problems as they are insensitive to large quantities of negative points. However, our context admits extra structure: a spatial component of our data points, not present in the feature space.

Consider the situation of a component with two problematic regions, one significantly larger than the other. If we were to naively apply a combination of precision and recall to points sampled from our object, we could obtain good performance by correctly labelling all points in the large region and ignoring the small region. However, from the point of view of the end-user, these problematic regions could be just as important as each other. The following definitions address this alternative source of imbalance.

Let x_i denote a set of points and y_i denote whether the point is printable or problematic. For a given predictive model, let $\widehat{y_i}$ denote the prediction of the printability of the point. For a fixed radius parameter r, define the *true* and *predicted problematic regions* by

$$\mathcal{B}_r = \partial T \cap \bigcup_{i \,:\, y_i = 1} B_r(x_i) \tag{10}$$

$$\widehat{\mathcal{B}}_r = \partial T \cap \bigcup_{i \,:\, \widehat{y_i} = 1} B_r(x_i) \tag{11}$$

respectively. We now define the *spatial* true positives, false negatives and false positives by

$$\mathbf{tp} = |\mathcal{B}_r \cap \widehat{\mathcal{B}}_r|, \quad \mathbf{fn} = |\mathcal{B}_r - \widehat{\mathcal{B}}_r| \quad \text{and} \quad \mathbf{fp} = |\widehat{\mathcal{B}}_r - \mathcal{B}_r|.$$

where $| - |$ denotes the area of the region. Armed with these definitions we define the *spatial recall*, *spatial precision* and *spatial* F_β in the obvious fashion. In practice we can estimate these values by counting the number of points in each region rather than computing the area.

The question remains of which subset of our data we should use to evaluate this metric. Typically, performance is assessed using a hold-out validation set, or by using a cross-validation scheme. In our context, we have a high degree of *spatial autocorrelation*: nearby points look similarly geometrically and have similar printability measures. Thus, to avoid leakage of information from our training set to our validation set we must be careful how we select our validation sets. Our solution is to select spatially disjoint validation sets, such as taking points lying in horizontal slices of our test artefact.

Results

Test artefact

Computer aided design We used the process detailed in "Data generation and processing" section, with translations uniformly sampled from a 100 mm × 100 mm × 10 mm cuboid to generate a random polyhedron. To help keep track of the orientation we added $1 \, \text{cm}^3$ cubes at 3 of the bounding box vertices. The resulting CAD had a bounding box of 117 mm × 116 mm × 29 mm.

Build process The manufacturing of the test artefact was carried out using an electron beam melting (EBM) system, ARCAM S12. Loaded with Ti–6Al–4V prealloyed powder as feedstock material, the size distribution was reported to be 45–106µm under the batch 1250 specifications supplied by ARCAM Gothenburg Sweden.

Prior to EBM manufacturing, CAD drawings were converted into a stereolithography file in order to be adapted for processing. Two-dimensional representation in layers of 50 µm was performed using the ARCAM build assembler in order to be processed by the EBM control software. Standard ARCAM themes for 50 *rmu*m layer thickness were used, with high power, speed, and defocused beam during preheating together with lower power, lower speed and a more focused beam to melt the powder (Hernández-Nava et al. 2016). Upon build completion, the artefact was extracted from the build tank to further be cleaned using compressed air in a contained volume, a powder recovery system.

Scanning procedure Following manufacture, the geometry of the sample was analysed by X-ray Computed Tomography (CT) in the Henry Moseley X-ray Imaging facility at the University of Manchester. To obtain a suitable resolution, it was necessary to slice the build using electron discharge machining to obtain a sample of dimensions 12 cm × 3 cm × 3 cm. The sample was clamped in a rotating stage in a Nikon Metrology 225/320 kV Custom Bay machine. To avoid interference from the clamp mechanism only the upper half of the sample was imaged. However, by scanning the sample twice, and inverting the sample between scans, it was possible to image the entire volume.
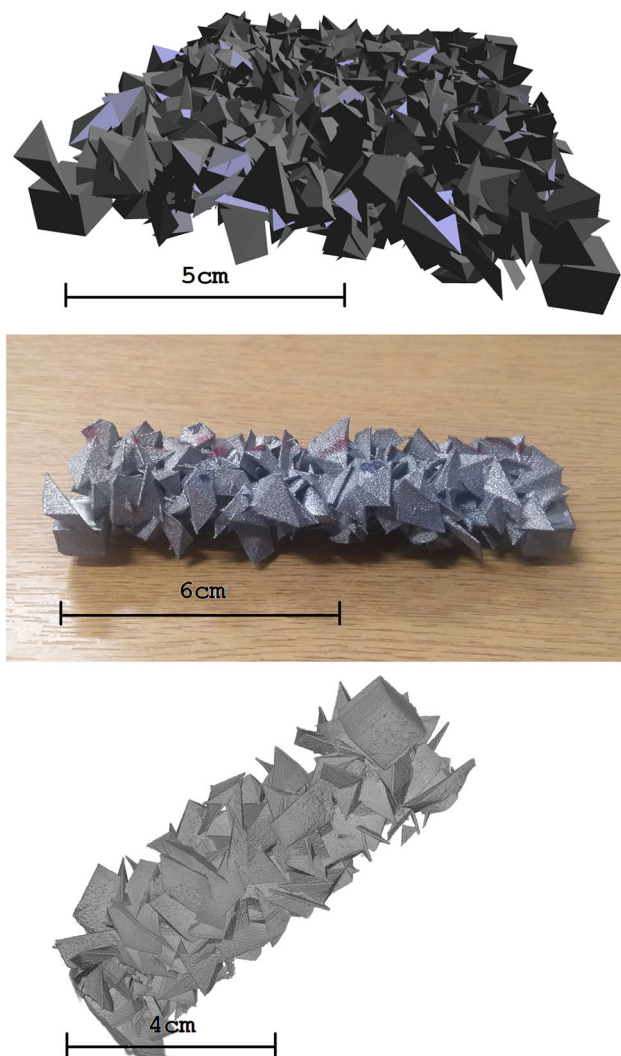
Fig. 6 The CAD, (slice of the) build and CT scan of our test artefact

imal (Landis and Keane 2010). The voxel size calculated automatically by the Nikon software was $40\,\mu$m. Given the machine used in this study was not calibrated for metrology, it is likely that the stage movement inaccuracies will result in a voxel size distinct from that calculated using the stated (uncalibrated) source to object and source to detector distance. However, in other studies (Villarraga-Gómez et al. 2018) comparisons between other metrology methods such as contact measurements, e.g. CMM (often assumed to have minimal uncertainty), and X-ray CT, have returned uncertainties similar to the difference between the Nikon provided voxel size and the compensated calculated voxel size observed in this work. In addition, the difference between CMM and CT measurements is often variable across the data set as found in (Villarraga-Gómez et al. 2018). We compensate for this uncertainty during the alignment procedure as detailed in "Data generation and processing" section. By visual inspection, we found using a value of $r = 10$ mm for the radius of regions to align locally worked well. A more in depth discussion of possible CT errors is provided by Maire and Withers (Maire and Withers 2014). Illustrations of the CAD, build and CT scan are shown in Fig. 6.

Vertex printability

We now turn our attention to our main question of interest, that of predicting printability. As discussed, both the threshold t for marking a point as printable or problematic and the value of β to optimise our F_β-metric are heavily application dependant and will be determined by business requirements. We fix $t = 0.5$ mm and $\beta = 2.0$ for all our predictive experiments. We chose t to distinguish genuine build defects from natural surface roughness due to the AM process, and β to encourage our model to detect problematic regions, at the expense of an increased number of false positives. We use 4-fold cross-validation to optimise our hyper-parameters according to the scheme discussed in "Model selection" section before evaluating our algorithm on the *test set*, consisting of the final 1/4 of the artefact and data not used during cross-validation.

Before studying the general question of printability, we first tackle the sub-problem of predicting the printability of the *independent vertices*: vertices v more than some distance d away from any face not containing v. As discussed in "Geometric descriptors" section, we have a complete invariant describing the local geometry: the spherical polygon associated to the vertex as well as an appropriate measure of similarity between two such polygons.

We build our predictive algorithm by taking the distance to mark a vertex as independent as $d = 1$ mm, and training a support vector machine on our similarity matrix. To combat our class imbalance, we weight each class according to the relative frequencies in the training data. We choose the reg-

The system was equipped with a 225 kV static multi-metal reflection anode source (Cu, Mo, Ag, and W) with a minimum focal spot size of $3\,\mu$m and a Perkin Elmer 2048 × 2048 pixels 16-bit amorphous silicon flat panel detector with $200\,\mu$m pixel size. An accelerating voltage of 160 kV and current of 110 mA was used to generate the X-ray beam. Prior to interacting with the sample the beam was filtered with a 1.5 mm copper source to remove low energy photons. An exposure time of 1.415 s was used to collect 2000 radiographs, with around 15% of the photons being absorbed when passing through the sample.

3D data was reconstructed from the 2D radiographs using a filtered back projection algorithm and proprietary Nikon software. The data was downsampled from 32 bit to 8 bit in Avizo 9.0 prior to analysis. However, given the large differences in absorption between the titanium test object and the surrounding air, the effect of the down-sampling on the accuracy of feature detection can be assumed to be min-

ularisation coefficient to maximise the mean cross-validated F_2-score on our training data.

To assess the performance of our algorithm we compare to two simple benchmark algorithms and a theoretical approximate upper bound of our performance. For the benchmark algorithms we consider a naive model, making random predictions according to ratio of problematic to printable points in the training data and a nearest neighbour model, which predicts the same class as the most similar point in the training data.

It is well known within the additive manufacturing industry, that printability results are not entirely replicable. Indeed, printing a strenuous component several times will see several different results. Thus, any classification algorithm will be subject to an upper bound of performance, due to the *Bayes error*: the smallest possible error rate due to the underlying stochasticity. After consulting industry experts, we believe that, due to the complexity of the part, this error rate will be relatively large. It is well known in the industry, that when a component significantly exceeds the limits of printability, the build process generates a lot of random noise due to swelling of components and from fragments being manipulated by the over-coater. Given the lack of a full reproducibility study, we attempt to estimate this limit via the following procedure.

For each vertex, we assume the printability is given by a random variable taking values of 'printable' or 'problematic'. To estimate the probability distributions of these random variables we study *cliques*: groups X of vertices with pairwise normalised overlap of at least 90%, i.e. $\kappa(x_i, x_j)^{\frac{1}{2}} \geq 0.9$ for all $x_i, x_j \in X$. For each clique X_i, we can estimate the probability distribution $C_i = \mathbb{P}(\rho_x | x \in X_i)$; the probability a vertex x from that clique is printable or problematic, denoted by the random variable ρ_x. Then, for an arbitrary vertex x with observed printability ρ_0, we estimate the true printability distribution via the formula

$$\mathbb{P}(\rho_x | \rho_0) = \sum_i \mathbb{P}(\rho_x | \rho_x \sim C_i) \mathbb{P}(\rho_x \sim C_i | \rho_0) \qquad (12)$$

where $\rho_x \sim C_i$ denotes ρ_x having probability distribution C_i.

An example of this procedure is detailed in Fig. 7. From our data, if we observe a vertex to be problematic, we estimate it to be problematic with probability 56.4% and if we observe a vertex to be printable, we estimate it to be printable with probability 97.1%. We now estimate the upper bound of performance of any algorithm by taking the observed printability classifications as our prediction and comparing to the values sampled according to these probabilities. This procedure estimates the score we would achieve if we were to build the artefact again and make predictions based on the values we observed originally.

If $\rho_0 \in \{+ \text{ (printable)}, - \text{ (problematic)}\}$ is the observed printability of the vertex x, then writing

$$C_i = (\mathbb{P}(\rho_x = +|x \in X_i), \mathbb{P}(\rho_x = -|x \in X_i))$$
$$P_{i|\rho_0} = (P_{i|+}, P_{i|-})$$
$$= (\mathbb{P}(\rho_x \sim C_i | \rho_0 = +), \mathbb{P}(\rho_x \sim C_i | \rho_0 = -))$$

for 4 cliques X_1, X_2, X_3, X_4 we have

$$\begin{array}{lll} X_1 = \{+, +\} & C_1 = (1, 0) & P_{1|\rho_0} = (\frac{2}{4}, 0) \\ X_2 = \{+, -\} & C_2 = (\frac{1}{2}, \frac{1}{2}) & P_{2|\rho_0} = (\frac{1}{4}, \frac{1}{6}) \\ X_3 = \{+, -, -\} \implies & C_3 = (\frac{1}{3}, \frac{2}{3}) & P_{3|\rho_0} = (\frac{1}{4}, \frac{2}{6}) \\ X_4 = \{-, -, -\} & C_4 = (0, 1) & P_{4|\rho_0} = (0, \frac{3}{6}) \end{array}$$

and therefore,

$$\mathbb{P}(\rho_x = +|\rho_0 = +) = \sum_i \mathbb{P}(\rho_x = +|\rho_x \sim C_i) \cdot P_{i|+}$$
$$= 1 \cdot \frac{2}{4} + \frac{1}{2} \cdot \frac{1}{4} + \frac{1}{3} \cdot \frac{1}{4} + 0 \cdot \frac{0}{4} = \frac{17}{24}$$
$$\mathbb{P}(\rho_x = -|\rho_0 = +) = 1 - \mathbb{P}(\rho_x = +|\rho_0 = +) = \frac{7}{24}$$
$$\mathbb{P}(\rho_x = +|\rho_0 = -) = \sum_i \mathbb{P}(\rho_x = +|\rho_x \sim C_i) \cdot P_{i|-}$$
$$= 1 \cdot \frac{0}{6} + \frac{1}{2} \cdot \frac{1}{6} + \frac{1}{3} \cdot \frac{2}{6} + 0 \cdot \frac{3}{6} = \frac{7}{36}$$
$$\mathbb{P}(\rho_x = -|\rho_0 = -) = 1 - \mathbb{P}(\rho_x = +|\rho_0 = -) = \frac{29}{36}$$

Fig. 7 Example of our method of estimating the Bayes' error

The performance of our predictive model and the benchmarks is shown in Fig. 8 on our 4 hold-out sets used in the cross-validation process and the additional test set. The confusion matrices for the deterministic algorithms (our support vector machine and the nearest neighbour benchmark) can be seen in Table 1. As we can see **the performance of our model compares favourably with the theoretical upper bound even on unseen data**. The theoretical upper bound is determined by the complexity of the build component. It seems that our test artefact is far beyond the geometric limits of printability and consequently we have introduced a lot of noise into the AM process.

General printability

We construct our general predictive model as illustrated in Fig. 1. We first predict problematic vertices using our support vector machine as above. For our non-vertex points, we construct a feature matrix consisting of two $7 \times 7 \times 7$ voxel invariants, at pitches 0.5 mm and 1 mm, each autoencoded to an 8-dimensional latent space together with our geometric invariants: curvature, roughness, overhang (each computed at radii 1 mm, 2 mm, 5 mm and 10 mm) and our sphere and ray thickness and reach measures. As some reach measures may be infinite, we truncate them at 5 mm. This provides us
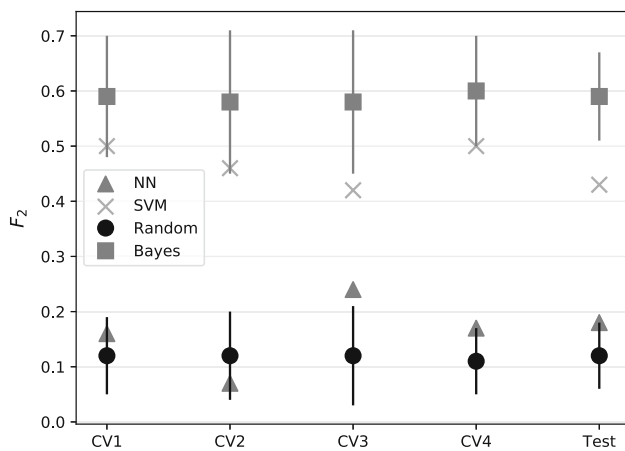
Fig. 8 F_2-scores for the random, nearest neighbour (NN), support vector machine (SVM) and estimated performance upper bound (Bayes) on the 4 cross-validation sets and our test artefact. For the non-deterministic algorithms, the error bars illustrate one standard deviation

Fig. 9 Spatial F_2-scores for the random, nearest neighbour (NN), our PCA, SMOTE, random forest model (Model) on the 4 cross-validation sets and our test artefact. For the non-deterministic algorithm, the error bars illustrate one standard deviation

Table 1 Confusion matrices for the two deterministic models evaluated on the test set

Predicted	Actual	
	Printable	Problematic
(a) Nearest Neigbour		
Printable	162	92
Problematic	24	8
Predicted	Actual	
	Printable	Problematic
(b) Support vector machine		
Printable	125	129
Problematic	8	24

with a 36-dimensional descriptor of the local geometry. As there is a large amount of correlation between several of our predictors, we apply PCA to reduce the dimension of our feature space. To combat the class imbalance problem, we apply the SMOTE technique and we use data augmentation to assist our model learning the invariance under the action of D_4 on our voxel invariants. Finally, we train a random forest on this resultant dataset to make our predictions. As before, we use 4-fold cross-validation to select the optimal combination of hyperparameters, according to our *spatial* metric of "Model selection" section with a radius of $r = 1$ mm.

Once again, we consider two benchmark algorithms as before. For the nearest neighbour algorithm, we first scale our features by subtracting the median and dividing by the inter-quartile range to robustly normalise across units and use the supremum norm to compute distances. The performance of our predictive model and the benchmarks is shown in Fig. 9. Once again, **we outperform our benchmarks**. Unfortunately, due to the lack of a suitable similarity mea-
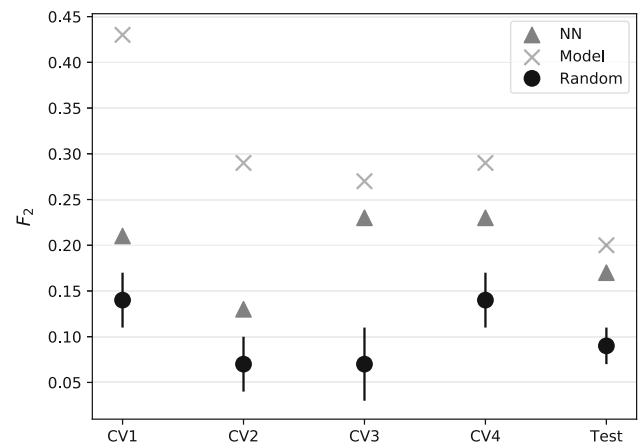
sure and issues stemming from spatial autocorrelation we are unable to estimate the theoretical upper bound of performance. However, we hope that future work, with a complete reproducibility study, will be able to confirm these results as close to the inherent limit.

Conclusions and further work

We have detailed a framework for predicting the printability of small-scale geometric features in additive manufacturing processes. This framework consists of numerous original components. Firstly, we provided an algorithm for constructing informative test artefacts which can be used to evaluate the geometric limits of additive manufacturing technologies. We detailed a method for measuring small-scale printability even on strenuous components containing large-scale defects. We provided several descriptors of local geometry which correlate with printability and are suitable inputs for many machine learning algorithms. Finally, we constructed predictive models which significantly outperform naive benchmarks and approach an estimate of the maximum performance obtainable due to inherent stochasticity in the underlying additive manufacturing process.

In further work, we intend to follow up with two key studies. First, we will test our predictive model on less stressful components with less inherent noise and see if we are still able to approach the (higher) theoretical performance limit. Secondly, we will test our predictive model on components from separate builds and analyse the effect of build-specific idiosyncrasies.

Another interesting avenue to explore is to study the printability as a regression problem. That is to say, attempt to quantify, up to uncertainty caused by the variability, the

printability $\rho(x)$ as a continuous variable. Our performance metrics of "Model selection" section, should have natural generalisations to the continuous setting following the work of Torgo and Ribeiro (2009). A similar avenue would be to attempt to measure the printability as a probability distribution and thus be able to concretely estimate the inherent noise in the additive manufacturing process.

To generalise this work, it would be useful to study the effects of different materials and process parameters on the printability. Given sufficient data, it may be possible to exploit similarities between materials and process parameters and be able to obtain a *universal algorithm* capable of taking these parameters as additional inputs.

Another direction for future work would be a data-driven study of the impact of geometric features on density. The density of the material in an additively manufactured component is not always constant throughout the part. Moreover, the density is closely related to material properties which directly impact the performance. It is well known in the industry that certain geometries are prone to having lower than optimal density. Thus, a potential direction for future research would be to replace the printability measure with a measure of the density of a region and retrain our models.

References

Aminzadeh, M., & Kurfess, T. (2018). Online quality inspection using Bayesian classification in powder-bed additive manufacturing from high-resolution visual camera images. *Journal of Intelligent Manufacturing*, *30*, 2505–2523. https://doi.org/10.1007/s10845-018-1412-0.

Azam, F. I., Rani, A. M. A., Altaf, K., Rao, T., & Zaharin, H. A. (2018). An in-depth review on direct additive manufacturing of metals. *IOP Conference Series: Materials Science and Engineering*, *328*(1), 012005.

Besl, P. J., & McKay, N. D. (1992). A method for registration of 3D shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *14*(2), 239–256. https://doi.org/10.1109/34.121791.

Bishop, C. (2013). *Pattern recognition and machine learning*. New Delhi: Springer.

Bowyer, K. W., Chawla, N. V., Hall, L. O., & Kegelmeyer, W. P. (2011). SMOTE: Synthetic minority over-sampling technique. CoRR arXiv:1106.1813.

Chazelle, B. M. (1981). Convex decompositions of polyhedra. In *Proceedings of the thirteenth annual ACM symposium on Theory of computing—STOC '81* (pp. 70–79). https://doi.org/10.1145/800076.802459.

Chen, X., & Schmitt, F. (1992). Intrinsic surface properties from surface triangulation. In *Proceedings of the Second European Conference on Computer Vision, ECCV '92* (pp. 739–743). Berlin, Heidelberg: Springer. http://dl.acm.org/citation.cfm?id=645305.648694.

Chen, Y., Garcia, E. K., Gupta, M. R., Rahimi, A., & Cazzanti, L. (2009). Similarity-based classification: Concepts and algorithms. *Journal of Machine Learning Research*, *10*, 747–776.

Cohen-Steiner, D., Morvan, & J. M. (2003). Restricted delaunay triangulations and normal cycle. In *Proceedings of the nineteenth annual symposium on computational geometry, SCG '03* (pp. 312–321). New York, NY: ACM. https://doi.org/10.1145/777792.777839.

Delgado, J., Ciurana, J., Reguant, C., & Cavallini, B. (2009) Studying the repeatability in DMLS technology using a complete geometry test part. In Innovative developments in design and manufacturing advanced research in virtual and rapid prototyping – Proceedings of VRP4, Leiria, Portugal.

Denlinger, E. R., & Michaleris, P. (2016). Effect of stress relaxation on distortion in additive manufacturing process modeling. *Additive Manufacturing*, *12*, 51–59. https://doi.org/10.1016/j.addma.2016.06.011.

Gao, H., & Ji, S. (2017). Efficient and invariant convolutional neural networks for dense prediction. CoRR arXiv:1711.09064.

Goodfellow, I. (2016). *Deep learning*. Cambridge, MA: The MIT Press.

Gower, J. C. (1975). Generalized procrustes analysis. *Psychometrika*, *40*(1), 33–51. https://doi.org/10.1007/BF02291478.

Hastie, T. (2009). *The elements of statistical learning: Data mining, inference, and prediction*. New York: Springer.

He, H., & Garcia, E. A. (2009). Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering*, *21*(9), 1263–1284. https://doi.org/10.1109/TKDE.2008.239.

Hernández-Nava, E., Smith, C., Derguti, F., Tammas-Williams, S., Leonard, F., Withers, P., et al. (2016). The effect of defects on the mechanical response of Ti–6Al–4V cubic lattice structures fabricated by electron beam melting. *Acta Materialia*, *108*, 279–292. https://doi.org/10.1016/j.actamat.2016.02.029.

Huang, J., Yagel, R., Filippov, V., & Kurzion, Y. (1998). An accurate method for voxelizing polygon meshes. In VVS '98: Proceedings of the 1998 IEEE symposium on volume visualization, pp. 119–126. https://doi.org/10.1145/288126.288181.

Inui, M., Umezu, N., & Shimane, R. (2016). Shrinking sphere: A parallel algorithm for computing the thickness of 3D objects. *Computer-Aided Design and Applications*, *13*(2), 199–207. https://doi.org/10.1080/16864360.2015.1084186.

Jee, H., & Witherell, P. (2017). A method for modularity in design rules for additive manufacturing. *Rapid Prototyping Journal*, *23*(6), 1107–1118. https://doi.org/10.1108/RPJ-02-2016-0016.

Jones, M. (1996). The production of volume data from triangular meshes using voxelisation. *Computer Graphics Forum*, *15*, 311–318.

King, W. E., Anderson, A. T., Ferencz, R. M., Hodge, N. E., Kamath, C., Khairallah, S. A., et al. (2015). Laser powder bed fusion additive manufacturing of metals; physics, computational, and materials challenges. *Applied Physics Reviews*, 2(4), 041304. https://doi.org/10.1063/1.4937809.

Kruth, J. P., Vandenbroucke, B., Van Vaerenbergh, J., & Mercelis, P. (2005). Benchmarking of different SLS/SLM processes as rapid manufacturing techniques, Vol. 10.

Kwon, O., Kim, H. G., Ji Ham, M., Kim, W., Kim, G. H., Cho, J. H., et al. (2018). A deep neural network for classification of melt-pool images in metal additive manufacturing. *Journal of Intelligent Manufacturing*. https://doi.org/10.1007/s10845-018-1451-6.

Landis, E. N., & Keane, D. T. (2010). X-ray microtomography. *Materials Characterization*, 61(12), 1305–1316. https://doi.org/10.1016/j.matchar.2010.09.012.

Li, C., Liu, Z., Fang, X., & Guo, Y. (2018). Residual stress in metal additive manufacturing. *Procedia CIRP*, 71, 348–353. https://doi.org/10.1016/j.procir.2018.05.039. **(4th CIRP Conference on Surface Integrity (CSI 2018))**.

Libbrecht, M. W., & Noble, W. S. (2015). Machine learning applications in genetics and genomics. *Nature Reviews Genetics*, 16, 321–332. https://doi.org/10.1038/nrg3920.

Mahesh, M., Wong, Y., Fuh, J., & Loh, H. (2004). Benchmarking for comparative evaluation of RP systems and processes. *Rapid Prototyping Journal*, 10(2), 123–135. https://doi.org/10.1108/13552540410526999.

Maire, E., & Withers, P. J. (2014). Quantitative X-ray tomography. *International Materials Reviews*, 59(1), 1–43. https://doi.org/10.1179/1743280413Y.0000000023.

Mani, M., Witherell, P., & Jee, H. (2017). Design rules for additive manufacturing: A categorization, p. V001T02A035.

Monostori, L., Markus, A., Van Brussel, H., & Westkämpfer, E. (1996). Machine learning approaches to manufacturing. *CIRP Annals - Manufacturing Technology*, 45(2), 675–712.

Moylan, S., Cooke, A., Jurrens, K., Slotwinski, J., & Donmez, M. A. (2012). *A review of test artifacts for additive manufacturing*. https://doi.org/10.6028/NIST.IR.7858.

Moylan, S. P., Slotwinski, J. A., Cooke, A., Jurrens, K. K., & Donmez, M. A. (2014). An additive manufacturing test artifact. *Journal of research of the National Institute of Standards and Technology*, 119, 429.

Mueller, B. (2012). Additive manufacturing technologies—Rapid prototyping to direct digital manufacturing. *Assembly Automation* 32(2), null. https://doi.org/10.1108/aa.2012.03332baa.010.

Panda, B., Shankhwar, K., Garg, A., & M. Savalani, M. (2016). Evaluation of genetic programming-based models for simulating bead dimensions in wire and arc additive manufacturing. *Journal of Intelligent Manufacturing*. https://doi.org/10.1007/s10845-016-1282-2.

Panozzo, D., Puppo, E., & Rocca, L. (2010). Efficient multi-scale curvature and crease estimation. In *2nd international workshop on computer graphics, computer vision and mathematics, GraVisMa 2010—Workshop Proceedings*, pp. 9–16.

Patil, S., & Ravi, B. (2006). Voxel-based representation, display and thickness analysis of intricate shapes. https://ieeexplore.ieee.org/document/1604669.

Pham, D. T., & Afify, A. A. (2005). Machine-learning techniques and their applications in manufacturing. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, 219(5), 395–412. https://doi.org/10.1243/095440505X32274.

Rusinkiewicz, S. (2004). Estimating curvatures and their derivatives on triangle meshes. In *Proceedings of the 3D data processing, visualization, and transmission, 2nd international symposium, 3DPVT '04* (pp. 486–493). Washington, DC: IEEE Computer Society. https://doi.org/10.1109/3DPVT.2004.54.

Santos, E. C., Shiomi, M., Osakada, K., & Laoui, T. (2006). Rapid manufacturing of metal components by laser forming. *International Journal of Machine Tools and Manufacture*, 46(12), 1459–1468. https://doi.org/10.1016/j.ijmachtools.2005.09.005.

Schneider, R. (2008). Recent results on random polytopes. *Bollettino dell Unione Matematica Italiana*, 1, 17–40.

Schoinochoritis, B., Chantzis, D., & Salonitis, K. (2015). Simulation of metallic powder bed additive manufacturing processes with the finite element method: A critical review. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, 231(1), 96–117.

Szilvási-Nagy, M. (1986). Two algorithms for decomposing a polyhedron into convex parts. *Computer Graphics Forum*, 5, 197–201.

Torgo, L., & Ribeiro, R. (2009). Precision and recall for regression. In J. Gama, V. S. Costa, A. M. Jorge, & P. B. Brazdil (Eds.), *Discovery science* (pp. 332–346). Berlin Heidelberg: Springer.

Villarraga-Gómez, H., Lee, C., & Smith, S. T. (2018). Dimensional metrology with X-ray CT: A comparison with CMM measurements on internal features and compliant structures. *Precision Engineering*, 51, 291–307. https://doi.org/10.1016/j.precisioneng.2017.08.021.

Yezzi, A. J., & Prince, J. L. (2003). An eulerian pde approach for computing tissue thickness. *IEEE Transactions on Medical Imaging*, 22(10), 1332–1339. https://doi.org/10.1109/TMI.2003.817775.

Yicha, Z., Bernard, A., Harik, R., K. P., K. (2015). Build orientation optimization for multi-part production in additive manufacturing. *Journal of Intelligent Manufacturing*. https://doi.org/10.1007/s10845-015-1057-1.