



THE UNIVERSITY *of* EDINBURGH

Edinburgh Research Explorer

## Time-efficient Surrogate Models of Thermal Modeling in Laser Powder Bed Fusion

**Citation for published version:**

Li, X & Polydorides, N 2022, 'Time-efficient Surrogate Models of Thermal Modeling in Laser Powder Bed Fusion', *Additive Manufacturing*, vol. 59, 103122. <https://doi.org/10.1016/j.addma.2022.103122>

**Digital Object Identifier (DOI):**

[10.1016/j.addma.2022.103122](https://doi.org/10.1016/j.addma.2022.103122)

**Link:**

[Link to publication record in Edinburgh Research Explorer](#)

**Document Version:**

Publisher's PDF, also known as Version of record

**Published In:**

Additive Manufacturing

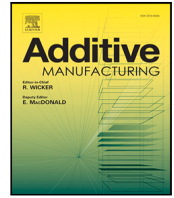
**General rights**

Copyright for the publications made accessible via the Edinburgh Research Explorer is retained by the author(s) and / or other copyright owners and it is a condition of accessing these publications that users recognise and abide by the legal requirements associated with these rights.

**Take down policy**

The University of Edinburgh has made every reasonable effort to ensure that Edinburgh Research Explorer content complies with UK legislation. If you believe that the public display of this file breaches copyright please contact [openaccess@ed.ac.uk](mailto:openaccess@ed.ac.uk) providing details, and we will remove access to the work immediately and investigate your claim.





Research paper

# Time-efficient surrogate models of thermal modeling in laser powder bed fusion

Xiaohan Li<sup>\*</sup>, Nick Polydorides

School of Engineering, University of Edinburgh, King's Buildings Campus, W Mains Rd, Edinburgh, EH9 3JW, UK

## ARTICLE INFO

## Keywords:

Finite element method  
Gaussian process emulator  
Model order reduction  
Randomized sketching  
Transient thermal model

## ABSTRACT

Two time-efficient surrogate models are proposed to emulate the nonlinear heat equation in the context of laser powder bed fusion, the performance of which is compared in accuracy and online execution time. Fast-computed numerical solvers are critical in implementing the digital twin framework in the additive manufacturing process addressing one of its main open problems: lack of quality assurance. The first surrogate model is the reduced Gaussian process emulator. It is a data-driven model equipped with a nonlinear dimension reduction scheme and manages to predict temperature profiles almost instantly (around 0.036s on average) with an accuracy of 95% for 99.38% of tests. Another surrogate model is the sketched emulator with local projection. It projects the accurate but high-dimensional finite element method solution on a low-dimensional basis and then bypasses the majority of costly computations for the temperature-dependent matrices in the projected model by randomized sketching. It has higher accuracy (97.78% of tests with relative errors below 1%) while spending comparably more time online (around 42.23s on average). Although compared with the finite element model both surrogates promote time efficiency with some minor controlled compromise in accuracy, the reduced Gaussian process emulator enables real-time implementation while the sketched emulator with local projection offers comparably higher levels of accuracy. A series of numerical experiments are carried out, which assumes a three-layer printing process with a fixed laser beam trajectory using a small number of printing control parameters as inputs, namely the laser power, scan speed, and time coordinates. Both surrogates are also principally feasible in other thermal-driven additive manufacturing to obtain better quality assurance with techniques like uncertainty management and closed-loop control.

## 1. Introduction

Laser powder bed fusion (LPBF), as a prevalent additive manufacturing (AM) technique in metal fabrication, successively spreads and selectively melts thin layers of powder to fabricate physical parts from 3D geometrical designs [1]. Though it is superior in realizing complex or personalized design resulting in increased application in fields like aerospace [2], automotive [3], and medical industry [4], quality assurance has been a persistent problem that makes digital twins of LPBF that help to approach the “first-time-right” goal receiving more and more attention [5]. As a thermal-driven AM, the thermal model of LPBF is a critical step in predicting the part properties such as microstructure and residual stress [6–8], according to which we can improve the part quality by designs like process optimization and closed-loop control [9,10]. Extensive researches with experimental validation have focused on the numerical solver with finite element method (FEM) such as the thermal model validated by an experiment with AlSi10Mg powder and argon atmosphere in [11,12] and the

nonlinear transient heat equation with phase changes and anisotropic thermal conductivity validated by the AM benchmark experimental set AMB2018-02 in [13]. Though the numerical simulators with FEM are precise, they are computationally expensive due to the high dimensionality caused by fine spatial–temporal discretization and the high nonlinearity caused by temperature-dependent thermal properties and boundary conditions. The high nonlinearity, however, means that we need to solve several high-dimensional linear equations to get the converged temperature result at a single time instant [14]. The expensive time cost detrimentally hinders its application both before and during printing processes, thus a swift but accurate heat transfer modeling of LPBF becomes a valuable research area.

To address the challenge of time consumption, various approaches have been proposed. One type of approach is purely data-driven. In [15], a recurrent neural network was trained to predict thermal profiles in the directed energy deposition process from different geometries. However, it required 250000 data points taking extensive

<sup>\*</sup> Corresponding author.

E-mail address: [xiaohan.li@ed.ac.uk](mailto:xiaohan.li@ed.ac.uk) (X. Li).

time and storage in data preparation and training. Another type of approach focus on model order reduction (MOR) since the thermal model of LPBF is a large-scale full order model (FOM) with significant redundancy. MOR can be assorted in three types: structural, data-driven, and projection-based approaches. Structural approaches refer to adaptive meshes [12,16,17] and substructure coupling [18,19]. In [16], an adaptive mesh refinement strategy was suggested to refine the mesh around geometric components while having coarser mesh for other areas. Jensen et al. described the dynamic analysis with substructures coupling internal dynamic behaviors and fixed-interface normal modes [18]. The data-driven MOR uses machine learning algorithms to find and/or predict low-dimensional latent representations of high-dimensional training outputs. Nikolopoulos et al. integrated a convolution autoencoder and a feed-forward neural network as a surrogate of partial differential equations (PDEs) along the entire time history [20]. The projection-based MOR projects high-dimensional PDEs into their reduced form where projection bases are obtained by methods such as moment matching [21], balanced truncation [22], and principal orthogonal decomposition (POD) [23]. Though the abovementioned attempts could expedite a thermal simulation, some obstacles remain. Data-driven approaches of exhaustive mapping generally rely on the representative quality of training data and normally need a large amount of training data which might be prohibitively expensive to generate. Projection-based approaches are challenging in finding proper and effective bases, and require additional approximations for nonlinear models such as discrete empirical interpolation method (DEIM) [24]. Nevertheless, a way out of the expensive time cost in thermal modeling of LPBF is surrogate models. The so-called surrogate model (SM) also known as emulator makes use of data generated by a high-fidelity physics-based model (the FEM in our case) and then swiftly predicts online with given design points, thereby inexpensively emulating the simulation [25].

Each SM individually balances model accuracy and online execution time. In this paper, we propose two SMs emulating the FEM with reasonable but different trade-offs. The first SM, the reduced Gaussian process (GP) emulator, learns a low-dimensional representation of the relative distance between temperature profiles which is later extrapolated to the prediction of high-dimensional temperature. The second SM, the sketched emulator with local projection, is a speedy numerical solver with FEM. It uses the relative distance predictor in the first SM to intelligently subsample the training temperatures and form effective local projection bases. The projected model is then sketched to suppress the time cost due to nonlinearity. Both SMs are time-efficient, but their run times and prediction accuracy are affected in different ways. The reduced GP emulator predicts temperatures very fast (0.036 s on average) with 99.38% of tests having relative errors below 5%, while the sketched emulator with local projection manages to predict 97.78% of tests with relative error less than 1% in an average execution time 42.23 s saving more than 82% of time required by FEM (4.23 min on average). Since both surrogates exhibit very good performance, we either choose one of them or indeed combine them to tackle problems requiring large and/or rapid thermal simulations in LPBF.

### 1.1. Notation and paper organization

We express matrices in capital letters, vectors and continuous functions in small case letter. For a matrix  $X$ ,  $X_{ij}$  is the  $(i, j)$ th element, the  $i$ th column is denoted as  $X_{*i}$ , and  $X_{i*}$  denotes its  $i$ th row. For a vector  $x$ , the  $i$ th entry is denoted by  $x_i$ .

In the next section, we describe the nonlinear transient heat equation model as well as its boundary conditions pertaining to LPBF and then outline its high-fidelity numerical solution with FEM. The section afterward illustrates the methods of two proposed SMs. The method of reduced GP emulator is sequentially depicted as the subsampling based on relative distance predictions and the approximation of high-dimensional temperatures based on subsampled temperatures. The

method of the sketched emulator with local projection is depicted by three subsections which contain local projection using relative-distance-based subsampling, the approximation of temperature-dependent matrices with randomized sketching, and the reconstruction of high-dimensional temperatures after solving the projected and sketched model. Then, the section of results and discussion follows, which details the numerical experiments comparing the performance of the two SMs. The prediction accuracy is compared in temperature profiles, thermal histories, and melt pool sizes, while the time efficiency is compared by average execution time. In the end, we have the section containing conclusions of designed models.

## 2. Numerical thermal modeling of LPBF

During the LPBF process, as shown in the schematic of Fig. 1, a high power-density laser beam selectively melts and fuses a thin layer of powder along a preset scan path. The printed object is then formed by bonding adjacent tracks and layers as new layers of powder are successively superimposed. Its thermal modeling entails a heat equation over a vertically extended domain, the boundary conditions of which consider a Gaussian heat flux, convection and radiation heat loss, and a temperature-controlled platform. The powder bed as a multi-layer spatial domain of heat equation is discretized by adaptive tetrahedron meshes, which yields a numerical solver with FEM.

### 2.1. Governing equations

The heat transfer in a 3D computational domain  $\Omega \in \mathbb{R}^3$  is governed by the nonlinear heat equation in Eq. (1). It is nonlinear due to the dependence of material properties on temperature. Let  $u := u(x, t)$  be the temperature at spatial coordinate  $x := [x_1, x_2, x_3]$  in a 3D Cartesian frame, and temporal coordinate  $t \in [0, t_f]$ . If  $\rho$ ,  $c$ , and  $\bar{\kappa}$  are respectively the temperature-dependent density, specific heat capacity and thermal conductivity tensor of the material then the temperature in the domain is known to satisfy

$$\rho(x, t)c(x, t)\frac{\partial u(x, t)}{\partial t} - \nabla \cdot \bar{\kappa}(x, t)\nabla u(x, t) = 0, \quad (x, t) \text{ in } \Omega \times [0, t_f], \quad (1)$$

where the temperature dependence of the thermal properties is derived from material-specific experimental data. The thermal conductivity  $\bar{\kappa}$  is modeled as an anisotropic tensor field, and the latent heat effect during a solid-liquid phase change is introduced in terms of the model of specific heat capacity  $c$  [13]. Further details of the models of nonlinear thermal properties are included in Appendix A.

The heat is imparted to the domain by means of a laser beam, modeled as a moving Gaussian heat source [11]

$$q_h(x, t) = \frac{2\alpha P}{\pi\gamma^2} \exp\left(-\frac{2\|x - x^c(t)\|^2}{\gamma^2}\right), \quad x \text{ on } \Gamma_t, \quad (2)$$

acting on the top surface of the domain  $\Gamma_t$ . In Eq. (2),  $\alpha$  is the heat absorptivity,  $P$  is the laser power, and  $\gamma$  is the effective laser beam radius, all three assumed invariant in space and time. In the same equation  $x^c$  is the position of the center of the laser beam at time  $t$  as it moves with a constant scan speed  $v$ . The laser energy is mainly absorbed by the powder resulting in melt pools, while some of it escapes in the atmosphere as a heat loss  $q_l$  satisfying

$$\bar{\kappa}(x, t)\nabla u(x, t) \cdot \hat{n} = q_h(x, t) - q_l(x, t), \quad x \text{ on } \Gamma_t \cup \Gamma_s, \quad (3)$$

where  $\hat{n}$  is the outward unit normal on the top surface  $\Gamma_t$  and side boundary  $\Gamma_s$ , while the respective convection and radiation constituents in  $q_l = q_c + q_r$  are

$$q_c(x, t) = h(u(x, t) - u_a), \quad q_r(x, t) = \sigma_s \varepsilon (u(x, t)^4 - u_a^4), \quad x \text{ on } \Gamma_t \cup \Gamma_s, \quad (4)$$

with  $h > 0$  is the heat convection coefficient,  $\sigma_s$  is the Stefan-Boltzmann constant,  $\varepsilon$  is the emissivity, and  $u_a$  the ambient temperature [26,27]. To improve the final object's quality, such as mechanical properties,

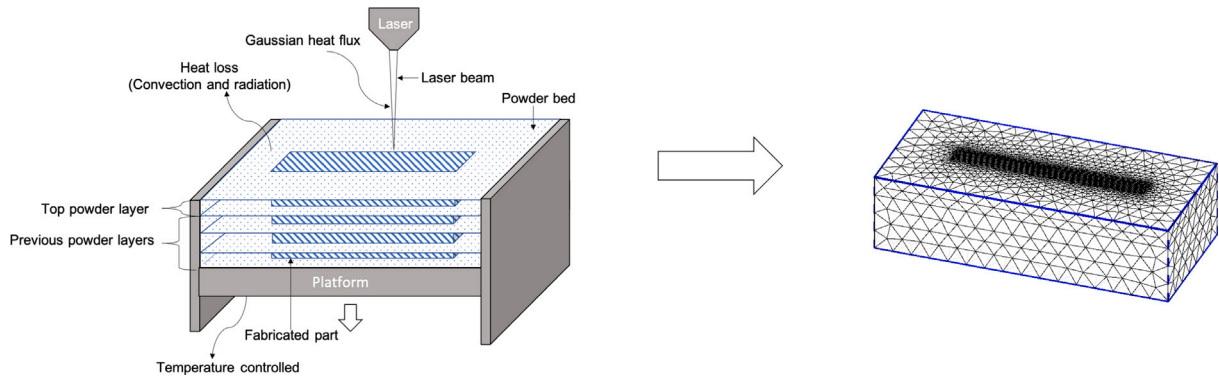


Fig. 1. The printing process of LPBF and its spatial discretization with FEM.

microstructure and residual stress, a temperature  $u_b$  is applied on the temperature-controlled bottom surface  $\Gamma_b$  hence a Dirichlet boundary condition

$$u(x, \cdot) = u_b, \quad x \text{ on } \Gamma_b, \quad (5)$$

is imposed at all times. Finally we assume the initial condition

$$u(\cdot, 0) = u_0, \quad x \text{ in } \Omega, \quad (6)$$

so that Eqs. (1)–(6) admit a unique solution  $u(x, t) \in \Omega \times [0, t_f]$ .

## 2.2. Finite element method

To solve the nonlinear equations as Eqs. (1)–(6) using the FEM, we respectively discretize the domain in space and time in linear tetrahedral elements and a number of time instants  $\{t_n \doteq n\Delta t\}_{n=0}^T$  for a given small  $\Delta t$ . The imposed boundary conditions and the temperature  $u_{n-1}$  yields a model with  $d$  degrees of freedom (DoF) for the temperature field  $u_n$  in the form of the nonlinear system

$$A_n(u_n)u_n = b_n(u_n), \quad n = 1, 2, \dots \quad (7)$$

where the temperature-dependent system matrix  $A_n$  and right hand side vector  $b_n$  at time  $t_n$  have dimension  $d$  which is typically very large. Details of the construction of these are given in Eqs. (B.1) and (B.2) in Appendix B.1, the matrices and vectors in which are defined as the integrals in Table B.1 and are numerically approximated with Gaussian quadrature rules [28,29] as in Table B.2 in Appendix B. The nonlinear heat equation as Eq. (7) is solved iteratively using the Picard algorithm [14] as we outline in Algorithm B.1 of Appendix B.4.

The three most time-consuming parts of the numerical solver with FEM are the high dimensionality caused by fine spatial and temporal discretization, the number of Picard iterations required to converge, and the computation of the FEM integrals involving high-order polynomials using Gaussian quadrature. To address these issues, two time-efficient SMs are proposed to significantly reduce the online execution time by exploiting offline data sets to minimize online computations.

## 3. Time-efficient surrogate models

The prevalent computational complexity associated with the high-dimensionality and non-linearity of FEM renders it unsuitable for real-time simulation. To decrease this computational complexity, we propose two time-efficient SMs: the reduced GP emulator and the sketched emulator with local projection. The reduced GP emulator first builds a fast-computed data-driven model of relative distances between temperature profiles and then extrapolates the relative distance predictions as high-dimensional temperature predictions. In the sketched emulator with local projection, we propose low-dimensional local projections and randomized sketching to expedite FEM and form a SM. The local projection bases themselves are generated with the help of the

relative distance predictor in the reduced GP emulator. Both SMs are significantly quicker than solving the FEM, and either of them may be used depending upon the desired trade-off between runtime and accuracy. Since a part of the reduced GP emulator is used to compute local projections for the sketched emulator, we begin this section by presenting the reduced GP emulator.

### 3.1. Reduced GP emulator

Data-driven approaches are frequently used in SMs where the result is obtained in near real-time as a direct prediction [20]. What hinders machine learning algorithms from emulating the thermal modeling in LPBF is the high dimensionality of thermal fields and the prohibitively expensive time in generating a large training data set. To address these challenges, we propose a reduced GP emulator where a subsampling scheme based on relative distance predictions is designed with Gaussian process regression (GPR) and nonlinear MOR, and a swift predictor is designed to extrapolate subsampled training temperatures and distance predictions to high-dimensional temperatures. The framework of the reduced GP heat emulator is as Fig. 2 where the module of subsampling and high-dimensional temperature predictor are respectively elaborated in Sections 3.1.1 and 3.1.2. In addition, the subsampling module is also used in the module of local projection demonstrated in the framework of the sketched emulator with local projection as Fig. 3.

#### 3.1.1. Subsampling

The high-dimensional temperature of a given test input can be predicted by a linear combination of some training temperatures deemed as the closest ones to this prediction, in which the weights of the linear combination are inversely proportional to relative distances between the temperature prediction and the selected training temperatures. We thus develop a subsampling scheme to select training temperature snapshots based on a relative distance predictor mapping input parameters to relative distances.

As a data-driven method, its prediction accuracy relies on finding representative training data set which is generated by the heat simulator with FEM. The training inputs are selected from the controllable parameters of a LPBF process including laser power, beam size, scan speed, time, preheating temperature, and so on. In this paper, the inputs matrix  $X \in \mathbb{R}^{\ell \times N}$  contains  $N$  training points for a triplet ( $\ell = 3$ ) of laser power  $P$ , scan speed  $v$ , and time  $t$ . For each  $X_{*i} \doteq [P_i, v_i, t_i]^T$ , one can readily run the heat simulator with FEM as Eq. (7) by tuning the appropriate boundary conditions for times  $t = t_1, \dots, t_i$  until we obtain  $u(X_{*i})$ . Repeating the process for  $N \gg 3$  sampling points produces a corresponding temperature snapshots matrix  $U \in \mathbb{R}^{d \times N}$  where  $U_{*i} = u(X_{*i})$  and  $N$  is typically chosen to be smaller than  $d$ . To make the learned parameters in GPR scale-invariant, we normalize the input quantities in the rows of  $X$  to be in the  $[0, 1]$  range

$$\tilde{X}_{i*} = \frac{X_{i*} - \min(X_{i*})}{\max(X_{i*}) - \min(X_{i*})}, \quad \text{for } i = 1, \dots, \ell. \quad (8)$$

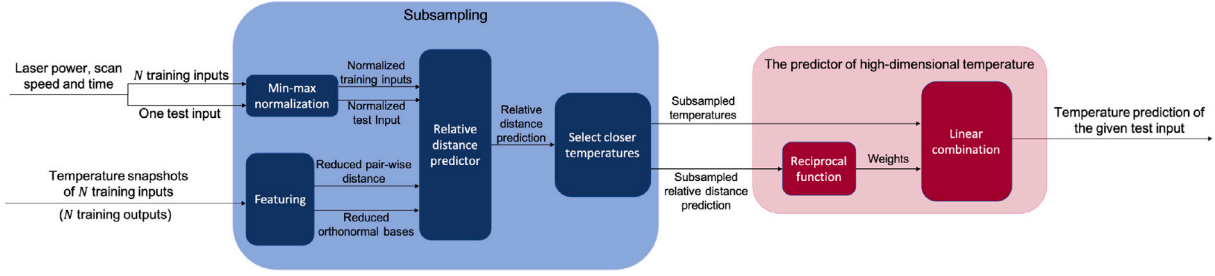


Fig. 2. The framework of the reduced GP emulator predicting high-dimensional temperatures with three control inputs (laser power, scan speed and time) via linear combinations of subsampled training temperatures.

As for the feature engineering of the training output, we approximate the pair-wise distances of temperatures in  $U$  with nonlinear dimensionality reduction [30]. The relative distance between  $U_{*i}$  and  $U_{*j}$  for  $i, j = 1, \dots, N$ , denoted as  $Q_{ij}$  in the pair-wise distance matrix  $Q \in \mathbb{R}^{N \times N}$ , is the shortest distance between the  $i$ th and  $j$ th node of a weighted  $g$ -neighborhood graph. The construction of this weighted graph and the computation of its shortest path distance are further detailed in appendix C. We then find a reduced representation of  $Q$  by a rank  $r$  truncation of its singular value decomposition (SVD) which is

$$Q = E\Lambda V^T \approx E_r Z, \quad (9)$$

where  $Z = A_r V_r^T$  is the  $r \times N$  reduced representation after featuring. Here,  $E \in \mathbb{R}^{N \times N}$ ,  $V \in \mathbb{R}^{N \times N}$ , and  $\Lambda \in \mathbb{R}^{N \times N}$  are respectively the left, right singular vectors, and the diagonal matrix of singular values.  $E_r = [E_{*1}, \dots, E_{*r}] \in \mathbb{R}^{N \times r}$ ,  $V_r = [V_{*1}, \dots, V_{*r}] \in \mathbb{R}^{N \times r}$ , and the diagonal matrix  $A_r \in \mathbb{R}^{r \times r}$  respectively contain the left, right singular vectors, and singular values corresponding to the  $r$  largest singular values in  $\Lambda$ . To guarantee the accuracy of reduction, the value of  $r$  is chosen to satisfy  $(\sum_{i=1}^r \Lambda_{ii}) / (\sum_{i=1}^N \Lambda_{ii}) > 99\%$ . After normalization and featuring, the resulting training data set is  $\{\tilde{X}_{*i}, Z_{*i}\}_{i=1}^N$  which represents a mapping from  $\ell$ -dimensional normalized inputs to  $r$ -dimensional reduced outputs with  $r \ll N$ .

As the preprocessed training inputs  $\tilde{X}$  and outputs  $Z$  are passed to the GPR module,  $r$  GPs are respectively trained to fit  $\{\tilde{X}, Z_{i*}\}_{i=1}^r$ . GP is promising in providing accurate predictions with corresponding variance using less lengthy training comparing with some other machine learning algorithms like neural networks. The multiple inputs and single output regression between  $\tilde{X} \in \mathbb{R}^{\ell \times N}$  and  $Y_{i*} = Z_{i*} - \frac{1}{N} Z_{i*} \mathbf{1}^T \in \mathbb{R}^{1 \times N}$  for  $i = 1, \dots, r$  is a discrete GP with zero mean and positive definite covariance matrix  $\Sigma^{(i)} \in \mathbb{R}^{N \times N}$ , which is

$$Y_{i*} \approx \mathcal{GP}(0, \Sigma^{(i)}), \quad (10)$$

where the covariance matrix  $\Sigma^{(i)}(\theta_{*i})$  arises from the discretization of a covariance function (kernel)  $k(x, x'; \theta^{(i)})$  with hyper-parameters  $\theta^{(i)}$ . More specifically,

$$\Sigma^{(i)} = \tilde{\Sigma}^{(i)} + \epsilon_i^2 I, \quad (11)$$

where

$$\begin{aligned} \tilde{\Sigma}_{pq}^{(i)} &\approx k(\tilde{X}_{*p}, \tilde{X}_{*q}; \theta^{(i)}) \\ &\doteq \theta_0^{(i)} \exp\left(-\frac{1}{2} \sum_{j=1}^{\ell} \frac{(\tilde{X}_{jp} - \tilde{X}_{jq})^2}{\theta_j^{(i)}}\right), \quad \text{for } p, q = 1, \dots, N, \end{aligned} \quad (12)$$

outlining the smoothness of GP as a squared exponential function of  $\tilde{X}$  [31]. The strictly positive hyper-parameters  $\theta_{*i} = [\theta_0^{(i)}, \theta_1^{(i)}, \dots, \theta_{\ell}^{(i)}, \epsilon_i]^{(i)}$  is optimally found by the maximum likelihood estimation. We hereafter respectively denote the  $i$ th column of the strictly positive matrix  $\hat{\theta} \in \mathbb{R}^{(\ell+2) \times r}$  and the covariance matrix  $\hat{\Sigma}^{(i)}$  as the trained hyper-parameters and the trained covariance matrix of the  $i$ th GP for  $i = 1, \dots, r$ . For a given test input  $x \in \mathbb{R}^{\ell}$  that is not in the training

data set, we first normalized it as  $\tilde{x}$ . Then, the prediction of its reduced representation  $\hat{z} \in \mathbb{R}^r$  and the corresponding variances  $\text{Var}(\hat{z})$  are

$$\hat{z}_i = B_{*i}^T S_{*i} + \frac{1}{N} Z_{i*} \mathbf{1}, \quad (13)$$

$$\text{Var}(\hat{z}_i) = \hat{\theta}_{1i} - B_{*i}^T (\hat{\Sigma}^{(i)})^{-1} B_{*i}, \quad \text{for } i = 1, \dots, r,$$

where the matrix  $B \in \mathbb{R}^{N \times r}$  and the matrix  $S \in \mathbb{R}^{N \times r}$  are respectively

$$B_{*i} = [k(\tilde{X}_{*1}, \tilde{x}; \hat{\theta}^{(i)}), \dots, k(\tilde{X}_{*N}, \tilde{x}; \hat{\theta}^{(i)})]^T, \quad (14)$$

$$S_{*i} = (\hat{\Sigma}^{(i)})^{-1} Y_{i*}^T, \quad \text{for } i = 1, \dots, r.$$

It is worth noting that only the matrix  $B$  is necessary to be computed online as it depends on the normalized test input  $\tilde{x}$ , while the more computationally expensive matrix  $S$  can be pre-computed offline. Therefore, the prediction of  $\hat{z}$  as Eq. (13) can be implemented instantaneously. We then reconstruct the full relative distance  $\hat{m} \in \mathbb{R}^N$  as

$$\hat{m} = E_r \hat{z}, \quad (15)$$

where  $E_r$  is the  $r$ -dimensional orthonormal bases in Eq. (9), and  $\hat{m}_i$  is the relative distance prediction between the temperature  $u(x)$  and the  $i$ th training temperature  $U_{*i}$ . In other words, the smaller  $\hat{m}_i$  is the closer the training temperature  $U_{*i}$  might be. We thereby could select a subset of training temperatures that are deemed to be more similar to the final temperature prediction  $u(x)$ .

### 3.1.2. High-dimensional temperature prediction

According to the prediction of relative distances  $\hat{m}$ , we select  $\bar{r}$  closest training data in  $U$  as  $\bar{U} \in \mathbb{R}^{d \times \bar{r}}$  where  $\bar{r}$  is normally selected as a comparably small value (like 10 or 20). To extrapolate the high-dimensional temperature we compute the weights  $w \in \mathbb{R}^{\bar{r}}$  of  $\bar{U}$  as

$$w_i \propto \frac{1}{\hat{m}_i}, \quad \text{such that } \sum_{i=1}^{\bar{r}} w_i = 1, \quad (16)$$

where the  $i$ th column  $\bar{U}_{*i}$  weighted by  $w_i$  is selected from the  $j$ th column in the original matrix of training temperatures  $U$ , and eventually the temperature prediction of the test input  $x$  is

$$\hat{u}(x) = \bar{U} w. \quad (17)$$

### 3.2. Sketched emulator with local projection

While the reduced GP emulator is entirely data-driven, the sketched emulator with local projection preserves the structure of the original heat simulator with FEM as Eq. (7) but expedites it by local projection and randomized sketching. Local projection targets at addressing the issue of high dimensionality with the help of subsampling described in Section 3.1.1. Randomized sketching, on the other hand, bypasses the majority of burdensome computation caused by high nonlinearity. The framework of the sketched emulator with local projection is as Fig. 3 where the module of local projection, randomized sketching and the high-dimensional temperature predictor are respectively explained in Sections 3.2.1, 3.2.2, and 3.2.3.

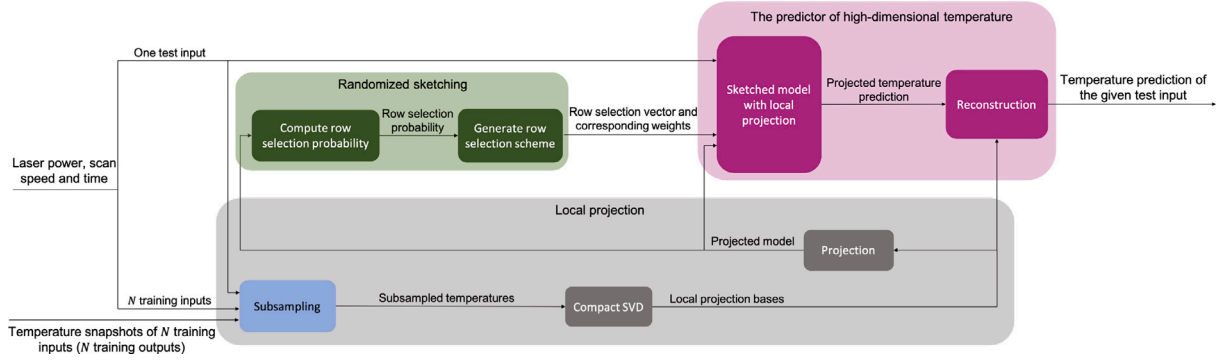


Fig. 3. The framework of the sketched emulator with local projection where high-dimensional temperatures of given test inputs (laser power, scan speed, and time) are reconstructed by the projected results produced from the emulator with subsampling-based local projection and randomized sketching.

### 3.2.1. Local projection

Recall the numerical solver with FEM as in Section 2.2 where the temperature at  $t_n = n\Delta t$  time instance is denoted as  $u_n \in \mathbb{R}^d$  on the discrete spatial domain with  $d$  DoF. Instead of directly computing Eq. (7), we can assume  $u_n \approx \Psi_n r_n$  for a low-dimensional vector  $r_n \in \mathbb{R}^{\tilde{r}}$  and project the FEM equations to the subspace spanned by the columns of  $\Psi_n$  to yield

$$\Psi_n^T A_n \Psi_n r_n = \Psi_n^T b_n, \quad (18)$$

where the projection basis  $\Psi_n \in \mathbb{R}^{d \times \tilde{r}}$  is orthonormal with  $\tilde{r} \ll d$ . The basis  $\Psi_n$  should span a space containing the final temperature result  $u_n$  with as less dimension  $\tilde{r}$  as possible. In the reduced GP emulator, we select a subset of training temperatures and weight them according to the relative distance predictor. It indicates that the selected subset forms a basis of the final temperature prediction. Therefore, we develop a similar subsampling scheme to generate local projection bases. Given a test input  $x_n = [P, v, t_n]^T$ , we predict its relative distance  $\hat{m}$  following Eqs. (13)–(15). Then, temperature snapshots in the training temperature matrix  $U$  corresponding to the  $\tilde{r}$  smallest values in  $\hat{m}$  are selected to form a subset  $\tilde{U} \in \mathbb{R}^{d \times \tilde{r}}$ . The local projection basis  $\Psi_n$  is the left singular vectors in the compact SVD of  $\tilde{U}$ . We hereafter use boldface to represent matrices and vectors after local projection such as  $\mathbf{A}_n := \Psi_n^T A_n \Psi_n \in \mathbb{R}^{\tilde{r} \times \tilde{r}}$  and  $\mathbf{b}_n := \Psi_n^T b_n \in \mathbb{R}^{\tilde{r}}$ . The discretized heat equation after local projection is rewritten as

$$\mathbf{A}_n r_n = \mathbf{b}_n. \quad (19)$$

One advantage of local projection is that it does not suffer from the curse of dimensionality as the number of inputs increases and/or each input has a larger range of interest. Both will cause a larger training data set (larger  $N$ ) yielding more resource cost in training and storage, but once we get the relative distance predictor the dimension after projection  $\tilde{r}$  remains small so the online execution time does not increase. However, the construction of local bases needs to be implemented online since in subsampling we use the values of test inputs which are not known offline, while the construction of global basis with POD method [23] can be finished offline since it constructs the global basis as the left singular vectors of the compact SVD of all temperature snapshots available thus independent from test inputs. In other words, local projection sacrifices the online time cost required to generate local project bases to further reduce the projected dimension from  $N$  to  $\tilde{r}$  with  $\tilde{r} \ll N$ .

Nonetheless, the projected model remains time-consuming due to its high nonlinearity. After local projection, the computation of nonlinear temperature-dependent functions in both  $\mathbf{A}$  and  $\mathbf{b}$  are not reduced. The number of nonlinear computations still depends on the original fine spatial discretization, the order of nonlinearity, and Gaussian quadrature rules. The difficulty of nonlinearity and its solution with randomized sketching are further elaborated in the following section.

### 3.2.2. Randomized sketching

The nonlinearity of the heat equation comes from temperature-dependent thermal properties and radiation heat loss, all of which are modeled as nonlinear functions. As stated in Appendices B.1 and B.2, the temperature-dependent parts of  $A_n$  and  $b_n$ , such as the stiffness matrix  $K_n \in \mathbb{R}^{d \times d}$ , the mass matrix  $M_n \in \mathbb{R}^{d \times d}$ , and the nonlinear part of radiation heat loss  $R_n \in \mathbb{R}^{d \times d}$ , need to integrate the corresponding nonlinear functions over the domain discretized into  $d$  nodes and  $e$  tetrahedral elements. For example, the mass matrix  $M_n$  is defined as

$$M_{n_{ij}} = \int_{\Omega} \rho(u_n) c(u_n) \phi_i \phi_j dx, \quad (20)$$

where  $\phi_i$  and  $\phi_j$  are respectively the basis function of the  $i$ th and  $j$ th node in FEM. The integration in Eq. (20) is approximated by taking  $n_m$  integration points in each element based on Gaussian quadrature rules [28,29], which is

$$M_n \approx \sum_{\tau=1}^{n_m} W_m^{(\tau)T} D_m^{(\tau)} W_m^{(\tau)}, \quad (21)$$

where  $W_m^{(\tau)} \in \mathbb{R}^{e \times d}$  is the matrix of basis functions in FEM evaluated at the  $\tau$ th integration point of each element, and  $D_m^{(\tau)} \in \mathbb{R}^{e \times e}$  is a diagonal matrix containing the product  $\rho(u_n)c(u_n)$  evaluated by the temperature at the  $\tau$ th integration point of each element. Further details of integral definitions and Gaussian quadrature approximations are listed in Appendices B.2 and B.3. After local projection, the projected mass matrix  $\mathbf{M}_n \in \mathbb{R}^{\tilde{r} \times \tilde{r}}$  becomes

$$\mathbf{M}_n = \sum_{\tau=1}^{n_m} \Psi_n^T W_m^{(\tau)T} D_m^{(\tau)} W_m^{(\tau)} \Psi_n = \sum_{\tau=1}^{n_m} \mathbf{W}_m^{(\tau)T} D_m^{(\tau)} \mathbf{W}_m^{(\tau)}, \quad (22)$$

where  $\mathbf{W}_m^{(\tau)} = W_m^{(\tau)} \Psi_n \in \mathbb{R}^{e \times \tilde{r}}$  for  $\tau = 1, \dots, n_m$  can be pre-computed since they are not temperature-dependent. The diagonals of  $D_m^{(\tau)}$  for  $\tau = 1, \dots, n_m$ , however, have a total of  $e \times n_m$  values awaiting for nonlinear computation of density and specific heat capacity at integration points in each Picard iteration. It is time-consuming as finer spatial discretization will increase  $e$  and higher order of nonlinearity will increase  $n_m$ . Accordingly, the expensive time cost due to nonlinearity is not reduced by a local projection which is also true for other temperature-dependent matrices like  $\mathbf{K}_n$  and  $\mathbf{R}_n$ . It is a major obstacle in implementing the projected model in real-time. To address this issue, we introduce randomized sketching based on Bernoulli sampling. Take the projected mass matrix  $\mathbf{M}_n$  as an example, the sum in Eq. (22) can be rewritten as the sum of  $n_m e$  sparse matrices which is

$$\mathbf{M}_n = \mathcal{W}_m^T D_m \mathcal{W}_m = \sum_{i=1}^{n_m e} D_{m_{ii}} \mathcal{W}_{m_{i\cdot}}^T \mathcal{W}_{m_{i\cdot}}, \quad (23)$$

where the thin matrix  $\mathcal{W}_m \in \mathbb{R}^{n_m e \times \tilde{r}}$  and the diagonal matrix  $D_m \in \mathbb{R}^{n_m e \times n_m e}$  are respectively

$$\mathcal{W}_m = \begin{bmatrix} \mathbf{W}_m^{(1)} \\ \vdots \\ \mathbf{W}_m^{(n_m)} \end{bmatrix}, \quad \text{and} \quad D_m = \begin{bmatrix} D_m^{(1)} & & \\ & \ddots & \\ & & D_m^{(n_m)} \end{bmatrix}. \quad (24)$$

We then sketch  $\mathbf{M}_n$  with only some of these sparse matrices weighted by a factor. Namely,

$$\hat{\mathbf{M}}_n = \sum_{i=1}^{n_m e} \xi_i \frac{D_{m_{ii}}}{\eta_i} \mathcal{W}_{m_{is}}^T \mathcal{W}_{m_{is}}, \quad (25)$$

where  $\xi_i$  is a Bernoulli random variable with probability of success  $0 \leq \eta_i \leq 1$ . We only manage to dramatically reduce evaluating  $D_{m_{ii}}$  without much accuracy compromise when many  $\xi_i = 0$  and  $\|\hat{\mathbf{M}}_n - \mathbf{M}_n\|$  is small with a high probability. To fulfill this, we set the probability  $\eta_i$  according to the thin matrix  $\mathcal{W}_m$  as Algorithm 1 [32]. The row selection scheme is accordingly generated as the row selection vector  $\xi_i$  and the corresponding weight  $1/\eta_i$  for  $i = 1, \dots, n_m e$ .

**Algorithm 1** The row selection and corresponding probability with randomized sketching

**Input:** a thin matrix  $\mathcal{W}_m \in \mathbb{R}^{n_m e \times \tilde{r}}$ , a constant  $\iota$ .

**Output:** a row selection vector  $\xi \in \mathbb{R}^{n_m e}$ , a probability vector  $\eta \in \mathbb{R}^{n_m e}$ .

- 1:  $\Phi \in \mathbb{R}^{n_m e \times \tilde{r}}$  is the left singular vectors in the compact SVD of  $\mathcal{W}_m$ .
- 2: **for**  $i = 1$  to  $n_m e$  **do**
- 3:  $\eta_i = \min\{1, \iota \|\Phi_{i*}\|^2\}$ .
- 4:  $\xi_i$  is generated as a Bernoulli random number with the success probability  $\eta_i$ .
- 5: **end for**

### 3.2.3. High-dimensional temperature prediction

Similar to the randomized sketching of the projected mass matrix  $\mathbf{M}_n$ , we can sketch other temperature-dependent matrices in the locally projected model. The sketched emulator with local projection then becomes

$$\hat{\mathbf{A}}_n \hat{r}_n = \hat{\mathbf{b}}_n. \quad (26)$$

After solving it with Picard iteration, we obtain an accurate result of  $\hat{r}_n$ . The prediction of high-dimensional temperature  $\hat{u}_n$  is then reconstructed by the local projection basis  $\Psi_n$  following

$$\hat{u}_n = \Psi_n \hat{r}_n. \quad (27)$$

## 4. Results and discussion

We develop a numerical experiment which is a printing process on a rectangular parallelepiped AlSi10Mg powder bed evolving from one to three layers in an argon atmosphere. The FEM-based numerical solver with fine spatial-temporal discretization, as a high-fidelity (ground-truth) heat simulation, is employed as a reference model. Its spatial domain is discretized with adaptive meshing as Fig. 4 to maintain the complexity to manageable levels as more layers are incorporated. While the printing area on the top layer keeps fine meshes being tetrahedrons with length 0.01 mm, the remaining meshes are gradually coarsened to tetrahedrons with length 0.1 mm. Under this adaptive mesh scheme, the DoFs in the one to three-layer domain are respectively  $d_1 = 13860$ ,  $d_2 = 13650$ , and  $d_3 = 14514$ . The scanning pattern plotted as red lines in Fig. 4 contains three straight lines scanned back and forth. The performance analysis is evaluated by model accuracy and online time cost compared between the reference simulator and the two SMs: the reduced GP emulator denoted as  $\mathcal{F}_1$  and the sketched emulator with local projection denoted as  $\mathcal{F}_2$ . All tests are run in Matlab R2020b on an Intel Core i5-9400F CPU at 2.90 GHz, 16 GB RAM computer.

### 4.1. Data generation

The thermal model with FEM following Eqs. (1)–(7) is validated with published experiments in [11] where LPBF processes with AlSi10Mg material in argon atmosphere were experimented and simulated. In [11], the computer-controlled selective laser melting process

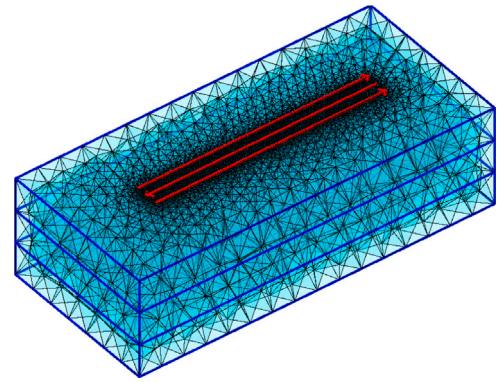


Fig. 4. The meshing and scanning pattern of the three-layer domain. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 1

Model parameters of the data generator [11,12,33].

Symbol	Definition (Unit)	Value
$\alpha$	Absorptivity	0.09
$\epsilon$	Emissivity	0.04
$\gamma$	Laser spot radius ( $\mu\text{m}$ )	35
$u_a$	Ambient temperature ( $^{\circ}\text{C}$ )	20
$u_b$	Building platform temperature ( $^{\circ}\text{C}$ )	200
$h$	Heat convection coefficient ( $\text{W}/(\text{m}^2\text{K})$ )	10
–	The size of one powder layer (mm)	$1.54 \times 0.7 \times 0.1$
$L$	Latent heat of AlSi10Mg ( $\text{kJ}/\text{kg}$ )	423
$\phi_p$	Powder bed porosity	0.4
$\sigma_s$	Stefan-Boltzmann constant ( $\text{W}/(\text{m}^2\text{K}^4)$ )	$5.67 \times 10^{-8}$

was conducted by the apparatus consisted of a YLR-500-SM ytterbium fiber laser and the spherical powder it used was 99.7% purity AlSi10Mg particles with an average size about  $30 \mu\text{m}$ . Its numerical simulation was taken by the ANSYS Multiphysics finite element package. With the parameters in [11], we validate temperatures generated by our thermal model in terms of highest temperature, maximum temperature gradient and melt pool size. The detailed comparison is listed as Tables D.1 and D.2 in Appendix D. The data generator is the thermal simulator with FEM. We hereby specify the model parameters as Table 1 and the thermal properties of AlSi10Mg and argon in Table A.1 in Appendix A. Sufficient training and testing data are essential in fulfilling the performance analysis. As discussed in previous sections the training input is of the form  $X_{*j} = [P_j, v_j, t_j]^T$  for  $j = 1, \dots, N$  containing laser power  $P$ , scan speed  $v$ , and time  $t$ . In the absence of the information on printing parameter distributions, we apply a uniform grid search in the admissible range of laser power  $P$ : 150–300 W with an interval of 15 W, scan speed  $v$ : 100–400 mm/s with an interval of 30 mm/s, and time  $t$ : the whole printing process with 30 time steps. With this sampling scheme, we respectively select 11, 11, and 30 different values for laser power, scan speed and time yielding 3630 ( $11 \times 11 \times 30$ ) pairs of data  $\{X_{*j}^{(i)}, U_{*j}^{(i)}\}_{j=1}^{3630}$  to construct the training data set of the  $i$ -layer domain  $X^{(i)} \in \mathbb{R}^{3 \times 3630}$  and  $U^{(i)} \in \mathbb{R}^{d_i \times 3630}$  for  $i = 1, 2, 3$ . The dimensionality of training temperatures  $d_i$ , however, is the number of DoF in the  $i$ -layer domain. Additionally, 9 simulations with different values of laser power (160, 220, 280) W and scan speed (115, 235, 355) mm/s are generated as the reference of tests, which yields 270 pairs of test inputs  $X_t^{(i)} \in \mathbb{R}^{3 \times 270}$  and the corresponding reference temperatures  $U_t^{(i)} \in \mathbb{R}^{d_i \times 270}$  in the  $i$ -layer domain with DoF  $d_i$  for  $i = 1, 2, 3$ .

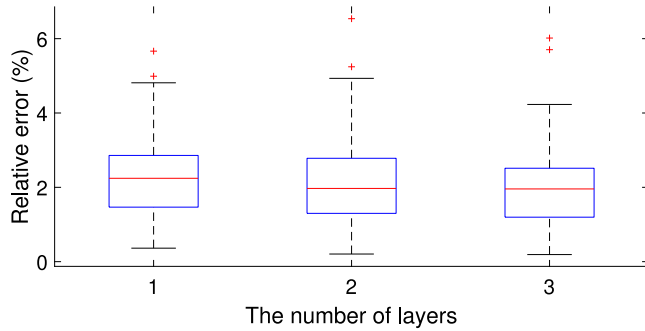
### 4.2. The performance of reduced GP emulator

#### 4.2.1. Dimension reduction

In the pre-processing of  $\mathcal{F}_1$ , the number of GP is reduced by a nonlinear dimension reduction technique. There are two important values in

**Table 2**  
The dimension reduction in  $\mathcal{F}_1$ .

Layer	Reduced dimension	Reduction percentage (%)
1	91	97.49
2	152	95.81
3	234	93.55



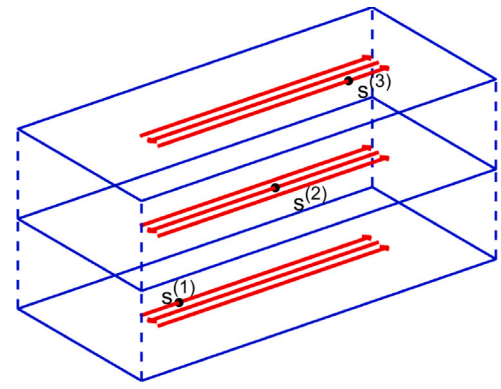
**Fig. 5.** The boxplot of relative errors between FEM and  $\mathcal{F}_1$ .

this method: one is the neighborhood size  $g$ , the value of which should keep the reachability between vertices in the weighted graph. In our case, we set  $g = 200$ . Another one is the target of dimension reduction  $r$ . It takes the minimum value that satisfies  $(\sum_{i=1}^r \Lambda_{ii}) / (\sum_{i=1}^N \Lambda_{ii}) > 99\%$  where  $\Lambda$  is the singular value diagonal matrix in Eq. (9). With the training temperature snapshots  $U^{(i)}$  for  $i = 1, 2, 3$ , the reduced dimension, as in Table 2, are respectively 91, 152, and 234. In comparison with the number of training data  $N = 3630$ , it is a dramatic reduction respectively cutting down 97.49%, 95.81%, and 93.55% of GPs required thus less effort in both training and prediction. It is also observed that the percentage of dimension reduction is decreasing as more layers are superimposed. One reasonable explanation is that the temperature and phase information of previous layers are kept, so the information is less redundant in the domain with more layers. Though this decreasing trend exists, it is not significant enough to have a detrimental effect on online time cost.

#### 4.2.2. Prediction accuracy

Given test inputs  $X_i^{(i)}$  in the  $i$ -layer domain for  $i = 1, 2, 3$ , we predict the corresponding temperatures as  $\hat{U}_i^{(i)} = \mathcal{F}_1(X_i^{(i)})$ . The prediction accuracy hereafter is evaluated by relative errors. For example, the relative error of the  $j$ th test in the  $i$ -layer domain is  $\|U_{t_{ij}}^{(i)} - \hat{U}_{t_{ij}}^{(i)}\| / \|U_{t_{ij}}^{(i)}\|$ . To evaluate the accuracy of all predictions, we graphically plot the spread of the relative errors of temperature profiles as the box plots in Fig. 5 where relative errors of all tests in the  $i$ -layer domain are plotted as the  $i$ th box. In a box, the 25th percentile ( $Q_1$ ), median ( $Q_2$ ), and 75th percentile ( $Q_3$ ) are respectively drawn as its bottom, red and top line. Two whiskers are additionally extended with a distance of  $1.5 \times (Q_3 - Q_1)$  where  $Q_3 - Q_1$  is also known as the interquartile range (IQR). Red crosses beyond whiskers represent outliers. We can tell from Fig. 5 that the majority of temperatures can be accurately predicted. Specifically, there are respectively around 17.78% and 99.38% of tests having relative errors below 1% and 5%, while the maximum relative error is around 6.54%.

Despite the relative error of temperature profiles, we also evaluate the prediction accuracy of thermal histories and melt pools. Take the printing process with laser power 280 W and scan speed 235 mm/s as an example, three positions selected from different scanned lines and different layers are used to compare the simulated and emulated results. The coordinates of these 3 positions are specified as  $s^{(1)} = (0.47, 0.40, 0.10)$  mm,  $s^{(2)} = (0.85, 0.35, 0.20)$  mm, and  $s^{(3)} = (1.13, 0.30, 0.30)$  mm, the locations of which are shown as Fig. 6. Thermal histories of  $s^{(i)}$  for  $i = 1, 2, 3$  throughout the three-layer printing



**Fig. 6.** The selected positions  $s^{(i)}$  for  $i = 1, 2, 3$  to compare thermal histories and melt pools.

are compared with the reference simulator in Fig. 7, the relative errors of which are respectively 4.49%, 4.51%, and 4.41%. The thermal history of one position starts from the layer it locates to the end of printing. In each layer of printing, three peaks are reached as the laser beam following the scan path of three straight lines approaches. The highest peak, however, appears when the laser beam passes right through the selected position yielding melt pools. It is also where obvious prediction discrepancy occurs while the general trend of curves is accurately approximated. The temperature distribution around melt pools at position  $s^{(i)}$  for  $i = 1, 2, 3$  are compared as Fig. 8 where the spatial distribution of relative error are also displayed. Higher relative errors concentrate around the areas with higher temperature, in which the maximums of relative errors are respectively 8.82%, 8.04%, and 8.38%. The highest temperature of melt pools at  $s^{(i)}$  for  $i = 1, 2, 3$  are respectively 2252.0 °C, 2308.2 °C, and 2462.8 °C in the FEM simulator, while with  $\mathcal{F}_1$  they are respectively emulated as 2390.7 °C, 2446.4 °C, and 2590.3 °C with relative error 6.16%, 5.99%, and 5.18%. Based on the comparison of the temperature colormaps around melt pools, it is indicated that the shape of melt pools is generally emulated. In specific, we compare the size of melt pools as Table 3 where the length, width, and depth of melt pool are respectively the maximum span of the melted area in  $x_1$ ,  $x_2$ , and  $x_3$  directions. At positions  $s^{(i)}$  for  $i = 1, 2, 3$ , the melt pool sizes are respectively predicted with relative error below 4.56%, 6.44%, and 3.10% but none of the emulated sizes is predicted with high precision ( $< 1\%$ ).

#### 4.2.3. Time cost reduction

With the level of accuracy compromise described above, a dramatic reduction of online time cost is accomplished since only swift computations are left for online execution. In specific, the comparison of time cost is listed as Table 4. Though the preparation time of data generation and training are required offline, the online execution time is respectively reduced from 3.88 min, 4.21 min, and 4.61 min to 0.024 s, 0.035 s, and 0.049 s on average in the one to three-layer domain. The dramatic reduction of time cost benefits from three advantages of the reduced GP emulator. First, the purely data-driven approach skips the burdensome computation caused by nonlinearity like Picard iteration and Gaussian quadratures. Second, only a comparably small number of GPs are required after nonlinear dimension reduction, which cuts the time cost of offline training and online prediction. Third, the prediction as Eq. (13) manages to simultaneously predict all GPs by executing the dot product between matrices instead of computing the univariate GP individually. While the performance in reducing online execution time is satisfactory, the compromise of accuracy is evident. Therefore,  $\mathcal{F}_1$  is suitable for applications that ask for real-time implementations but are endurable to some accuracy compromise.



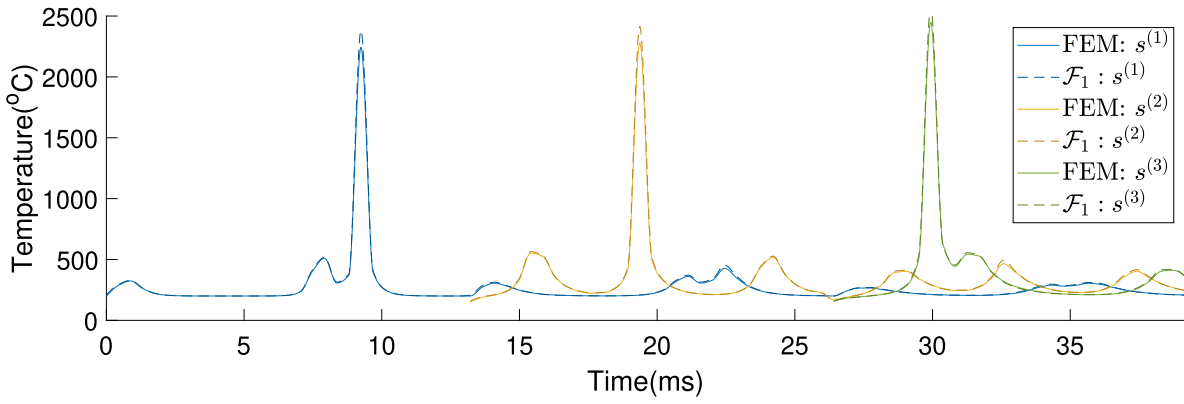


Fig. 7. The thermal history comparison between FEM and  $\mathcal{F}_1$ .

Table 3  
The melt pool size comparison between FEM and  $\mathcal{F}_1$ .

Size	$s^{(1)}$			$s^{(2)}$			$s^{(3)}$		
	Length	Width	Depth	Length	Width	Depth	Length	Width	Depth
FEM ( $\mu\text{m}$ )	132.78	93.76	66.94	162.58	100.75	80.80	192.14	118.92	93.24
$\mathcal{F}_1$ ( $\mu\text{m}$ )	138.84	96.74	68.97	171.69	103.74	86.00	197.63	122.60	95.86
Relative error (%)	4.56	3.17	3.03	5.60	2.97	6.44	2.86	3.10	2.81

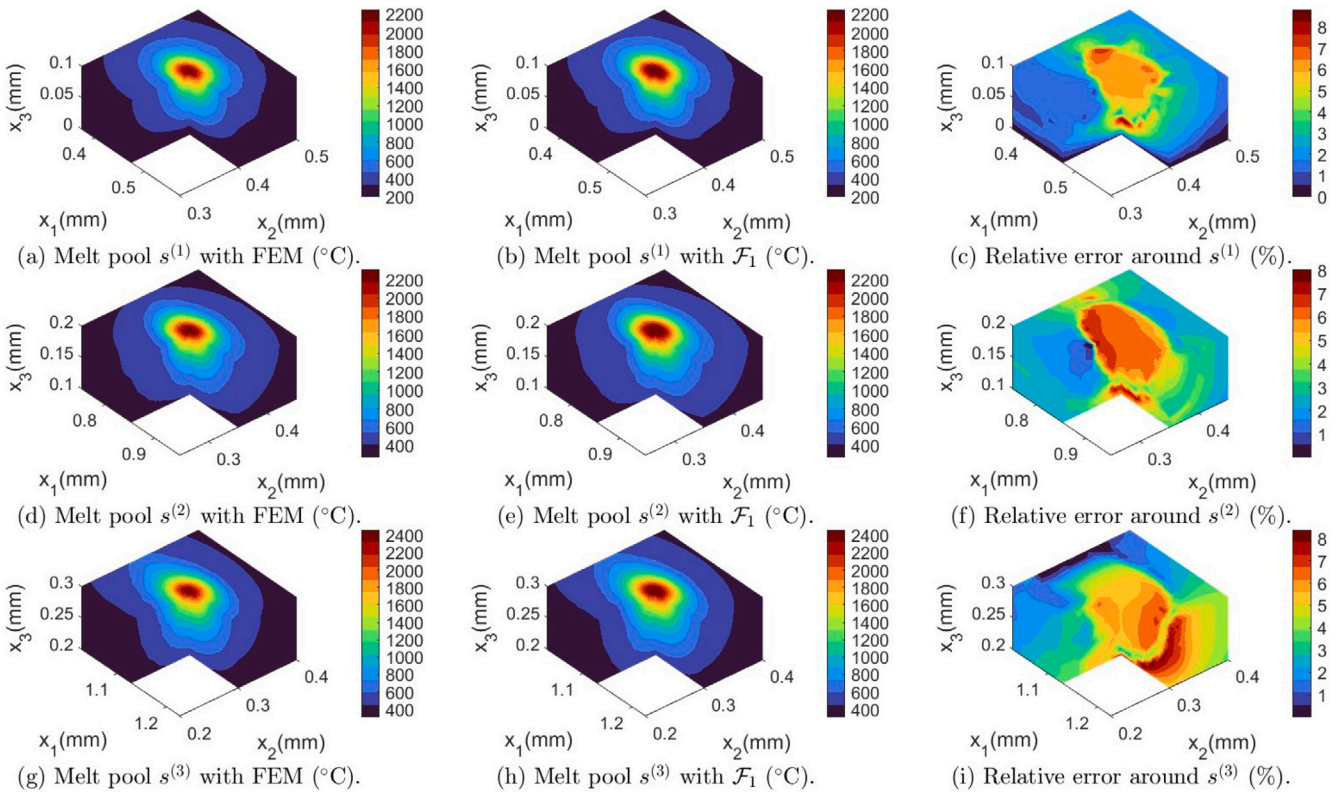
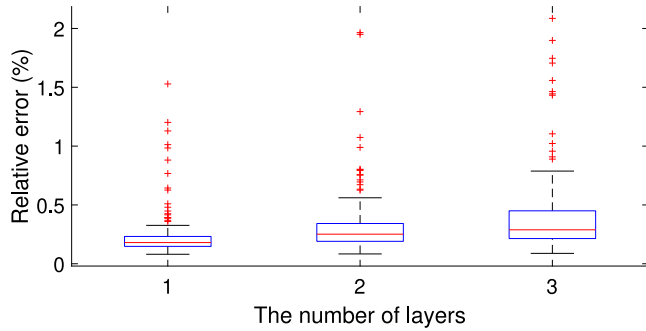


Fig. 8. The melt pool comparison between FEM and  $\mathcal{F}_1$ .

**Table 4**  
The time cost comparison between FEM and  $\mathcal{F}_1$ .

Layer	Data generation (day)	Training (h)	Average execution time	
			FEM (min)	Prediction (s)
1	12.74	16.71	3.88	0.024
2	15.74	27.69	4.21	0.035
3	16.53	40.07	4.61	0.049



**Fig. 9.** The relative errors between FEM and  $\mathcal{F}_2$ .

#### 4.3. The performance of sketched emulator with local projection

For the performance analysis of the SM  $\mathcal{F}_2$ , we develop temperature predictions with the same test inputs  $X_t^{(i)}$  for the convenience of comparison. For each pair of test input, specifically  $X_{t_{s_j}}^{(i)}$  for  $j = 1, \dots, 270$ , a local projection basis is formed by selecting 30 temperature snapshots from  $U^{(i)}$  that are predicted as the closest ones to  $U_{t_{s_j}}^{(i)}$ . After projecting and sketching, the temperature profiles are predicted as  $\hat{U}_t^{(i)} = \mathcal{F}_2(X_t^{(i)})$  for  $i = 1, 2, 3$ .

##### 4.3.1. Prediction accuracy

Relative errors between  $U_t^{(i)}$  and  $\hat{U}_t^{(i)}$  spread as the boxplot in Fig. 9, from which we can tell that  $\mathcal{F}_2$  is very accurate. 97.78% of relative errors are less than 1%, and the maximum relative error is 2.09%. We then use the same example (positions  $s^{(1)}$ ,  $s^{(2)}$ , and  $s^{(3)}$  in the thermal simulation with laser power  $P = 280$  W and scan speed  $v = 235$  mm/s) to show the accuracy of  $\mathcal{F}_2$  in emulating thermal histories and melt pools, which are respectively drawn in Figs. 10 and 11. At positions  $s^{(i)}$  for  $i = 1, 2, 3$ , the thermal histories are respectively predicted with relative errors 0.24%, 0.27%, and 0.25%. Throughout the entire printing process, the temperature curves are finely recovered indicating precise predictions of the peak temperature, heating and cooling rate. The spatial distribution of temperatures around melt pools at  $s^{(i)}$  for  $i = 1, 2, 3$  are delineated in Fig. 11 where relative errors of temperature prediction are respectively below 1.26%, 0.90%, and 1.12%. In comparison with FEM, the highest temperatures estimated as 2256.2 °C, 2310.0 °C, and 2462.9 °C with  $\mathcal{F}_2$  respectively have relative errors 0.19%, 0.08%, and 0.006%. Listed in Table 5, with  $\mathcal{F}_2$  the width and depth of  $s^{(1)}$ , the width of  $s^{(2)}$ , and the width of  $s^{(3)}$  are exactly emulated while the maximum relative error of other melt pool size predictions is 0.82%.

##### 4.3.2. Time cost reduction

Since the accuracy compromise is small, it is expected that  $\mathcal{F}_2$  needs more running time. While the preparation time of data generation and training is the same as  $\mathcal{F}_1$ , the total online execution time, listed as 41.50 s, 41.67 s, and 43.51 s on average for the one to three layers domain in Table 6, is composed of three parts: projection, sketching, and prediction. It is noted that in the  $i$ -layer domain for  $i = 1, 2, 3$  the average execution time of projection (2.17 s, 2.66 s, and 2.78 s) and sketching (6.19 s, 6.52 s, and 6.88 s) is much less

than prediction (33.14 s, 32.49 s, and 33.85 s) since projection and sketching only needs to be computed once with one set of test inputs while the prediction part includes the Picard iteration taking several runs to converge. As the number of layers increases, the execution time of projection, sketching, and prediction slightly increases but meanwhile the percentage of time cost reduction also increases. It means that compared with FEM the online running time of  $\mathcal{F}_2$  is less sensitive to more elements in the discretized domain. However, the online execution time is still observed with a significant reduction respectively saving 82.17%, 83.50%, and 84.27% of time cost in the one to three layers domain. The promising advantage of  $\mathcal{F}_2$  is that it expedites the high-fidelity simulator with a small compromise of accuracy.  $\mathcal{F}_2$  manages to make most predictions with relative error less than 1% using no more than 18% of online running time required by FEM, thus its performance is encouraging for applications that require accurate thermal simulations but do not excessively demand real-time implementations.

#### 4.4. Performance comparison

Two time-efficient SMs: the reduced GP emulator  $\mathcal{F}_1$  and the sketched emulator with local projection  $\mathcal{F}_2$  are compared with the same training data set  $\{X^{(i)}, U^{(i)}\}$  and test data set  $\{X_t^{(i)}, U_t^{(i)}\}$  in the  $i$ -layer domain for  $i = 1, 2, 3$ . Both SMs aim to reduce the online computation time while maintaining accuracy. However, they are, by design, different trade-offs between accuracy and time cost.  $\mathcal{F}_1$  manages to predict temperatures very fast (0.024 s, 0.035 s, and 0.049 s on average for one to three layers) with relatively higher prediction errors (17.78% and 99.38% of tests have relative errors less than 1% and 5%), while  $\mathcal{F}_2$  produces very accurate results (97.78% of tests have relative errors less than 1%) in a comparably longer execution time (41.50 s, 41.67 s, and 43.51 s on average for one to three layers). In other words,  $\mathcal{F}_1$  is biased on time cost reduction while  $\mathcal{F}_2$  is biased on prediction accuracy. As a data-driven method with nonlinear dimension reduction,  $\mathcal{F}_1$  highly relies on finding representative training data set to guarantee accuracy. This reliance also exists in  $\mathcal{F}_2$  since it uses the same relative distance predictor in  $\mathcal{F}_1$  to construct local projection bases. Nonetheless, the generator of local projection bases has less accuracy requirement in relative distance prediction as all subsampled temperature snapshots equally contribute to local projection bases, while in  $\mathcal{F}_1$  the exact value and order of relative distance prediction matters since they are used in computing proper weights to extrapolate the high-dimensional temperature predictions. This difference makes  $\mathcal{F}_2$  more robust in a different sampling of training data.

We can tell from the analysis above that both SMs have good performance in the reduction of time cost and the maintenance of accuracy. The less accurate  $\mathcal{F}_1$  still suppresses most prediction error less than 5% while the slower  $\mathcal{F}_2$  still manages to save more than 82% of time cost. We could select  $\mathcal{F}_1$  or  $\mathcal{F}_2$  based on the requirement of applications. If only a general description of temperature profiles is enough,  $\mathcal{F}_1$  is a more effective choice. Wherever precise analysis is necessary,  $\mathcal{F}_2$  is preferable when the execution time in dozens of seconds is acceptable.

## 5. Conclusions

Together with a nonlinear thermal model of LPBF using FEM, two time-efficient SMs: the reduced GP emulator and the sketched emulator with local projection are proposed. To display the performance of two SMs, a numerical experiment is displayed, which is a three-layer printing process scanning three straight lines back and forth with AlSi10Mg material and argon atmosphere. One SM is a reduced GP emulator which first predicts relative distance between temperature profiles with GPR and nonlinear dimension reduction and then extrapolates high-dimensional temperature predictions based on this prediction. It is implemented in real-time (the average execution time is 0.036 s)

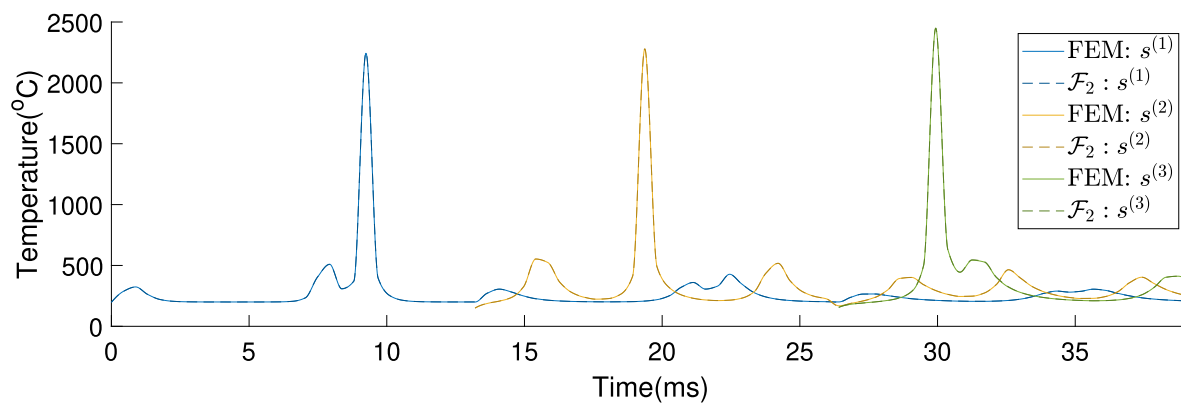


Fig. 10. The thermal history comparison between FEM and  $F_2$ .

Table 5  
The melt pool size comparison between FEM and  $F_2$ .

Position	$s^{(1)}$			$s^{(2)}$			$s^{(3)}$		
	Length	Width	Depth	Length	Width	Depth	Length	Width	Depth
$F_2$ ( $\mu\text{m}$ )	133.00	93.76	66.94	163.91	100.75	81.13	191.00	118.92	92.76
Relative error (%)	0.16	< 0.01	< 0.01	0.82	< 0.01	0.41	0.59	< 0.01	0.51

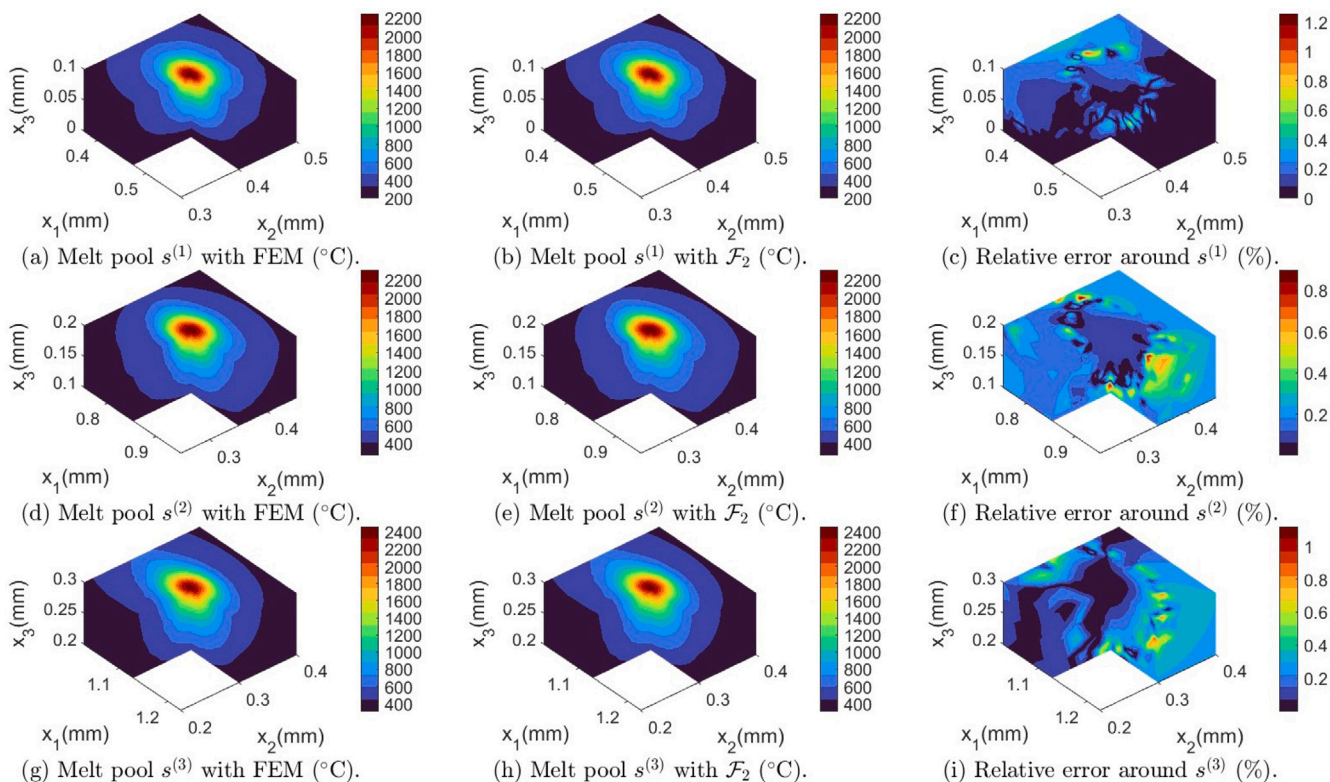


Fig. 11. The melt pool comparison between FEM and  $F_2$ .

**Table 6**  
The online time cost of  $F_2$ .

Layer	Average execution time				
	Projection (s)	Sketching (s)	Prediction (s)	Total (s)	Reduction (%)
1	2.17	6.19	33.14	41.50	82.17
2	2.66	6.52	32.49	41.67	83.50
3	2.78	6.88	33.85	43.51	84.27

while respectively keeping 17.78% and 99.38% of tests with relative errors below 1% and 5%. Another SM, a sketched emulator with local projection, is an expedited FEM solver. It first locally projects the FEM with bases formed by subsampled training data based on the relative distance prediction and then bypasses most nonlinear computation via randomized sketching. It is a very accurate SM (97.78% of tests with relative errors below 1%) while taking a comparably longer time online (the average execution time is 42.23 s). However, it still significantly reduces more than 82% of execution time compared with FEM. Despite the accuracy and time cost of temperature profile predictions, we also compare thermal histories and melt pools at selected positions. Both are finely emulated by the sketched emulator with local projection while being roughly recovered by the reduced GP emulator with visible deviation. The performance of the two SMs indicates that they are different trade-offs between accuracy and time cost: the reduced GP emulator outperforms in online execution time while the sketched emulator with local projection outperforms in model accuracy. In addition, though the data-driven relative distance predictor is used in both SMs, the subsampled training data based on relative distance predictions equally contribute in local bases of the sketched emulator while the exact values of relative distance predictions are used to weight the subsampled training data in the reduced GP emulator. This makes the sketched emulator with local projection less reliant on finding representative training data since it is more robust to the prediction error of relative distances. Overall, it is reasonable to select one SM based on the demand in precision and time cost for different applications. Both provide fast computations of temperature profiles, thermal histories, and melt pool sizes, the values of which are necessary for the models of microstructure and residual stress yielding the prediction of the final part's mechanical properties, deformation, and fatigue life. They, therefore, take one step further in facilitating the quality assurance of final parts. Moreover, though these two SMs are proposed for LPBF, they are essentially applicable for other thermal-driven AM by altering domain shape, heat source model, and so on. The future work will focus on how to properly apply these two SMs in the thermal analysis of LPBF such as defects prediction, uncertainty management, parameter optimization, and closed-loop controls.

#### CRedit authorship contribution statement

**Xiaohan Li:** Writing – original draft, Visualization, Validation, Software, Methodology, Formal analysis, Data curation. **Nick Polydorides:** Writing – review & editing, Supervision, Methodology, Investigation, Conceptualization.

#### Declaration of competing interest

One or more of the authors of this paper have disclosed potential or pertinent conflicts of interest, which may include receipt of payment, either direct or indirect, institutional support, or association with an entity in the biomedical field which may be perceived to have potential conflict of interest with this work. For full disclosure statements refer to <https://doi.org/10.1016/j.addma.2022.103122>. Xiaohan Li reports financial support was provided by The University of Edinburgh School of Engineering.

#### Data availability

Data will be made available on request.

#### Acknowledgments

We are grateful to the Principal's Career Development and Edinburgh Global Research Scholarship from the School of Engineering at the University of Edinburgh, UK for sponsoring this research. For the purpose of open access, the author has applied a Creative Commons Attribution (CC BY) license to any Author Accepted Manuscript version arising from this submission.

#### Appendix A. Nonlinear thermal properties

##### A.1. Latent heat

The latent heat effect during the phase change between solid and liquid is characterized in the model of specific heat capacity  $c$ , which is

$$c = \bar{c} + L \frac{\partial f_p}{\partial u}, \quad (\text{A.1})$$

where  $L$  is the specific latent heat of materials.  $f_p = 1/(1 + \exp(-\beta(u - (u_s + u_l)/2)))$  is a temperature-dependent sigmoid function, in which  $\beta > 0$  is the logistic growth rate controlling smoothness.  $u_s$  and  $u_l$  are respectively the lower and upper bound of mushy area [13].

##### A.2. Anisotropy of thermal conductivity

The anisotropic enhanced thermal conductivity model of thermal conductivity means that the thermal conductivity in the liquid phase is proportionally different along with the three mutually perpendicular directions in  $x := [x_1, x_2, x_3]$ . In details, the elliptic term in Eq. (1) becomes

$$\nabla \cdot \bar{\kappa} \nabla u := \sum_{i,j=1}^3 \frac{\partial}{\partial x_i} \left[ \bar{\kappa}_{ij} \right] \frac{\partial u}{\partial x_j} \quad (\text{A.2})$$

where  $\bar{\kappa}_{ij} = 0$  if  $i \neq j$ , and  $\bar{\kappa}_{11} : \bar{\kappa}_{22} : \bar{\kappa}_{33} = \lambda_1 : \lambda_2 : \lambda_3$ . The three-entry vector  $\lambda := [\lambda_1, \lambda_2, \lambda_3]^T$  scales the anisotropic thermal conductivity in liquid phase [13].

##### A.3. Polynomial fitting

The temperature-dependent thermal properties including thermal conductivity  $\bar{\kappa}$ , density  $\rho$ , and specific heat capacity  $\bar{c}$  are modeled by fitting experimental data. These thermal properties are modeled as piecewise polynomial functions in three temperature ranges:  $u \leq u_s$ ,  $u_s < u < u_l$ , and  $u \geq u_l$ , which respectively indicates the solid phase, the mushy area, and the liquid phase [11]. The fitted polynomials of AlSi10Mg and argon are detailed in Table A.1.

##### A.4. Thermal properties of powder

For the temperature range  $u \leq u_s$ , the materials have two phases in LPBF: powder and solid. Different from thermal properties of solid which are fitted polynomials based on experimental data, thermal properties of powder denoted as  $\bar{\kappa}_p$ ,  $\rho_p$  and  $\bar{c}_p$  in Eq. (A.3) are modeled according to the porosity of powder bed  $\phi_p$ , the materials, and the inert gas atmosphere [34,35].

$$\bar{\kappa}_p = (1 - \phi_p)^{1.5} \bar{\kappa}, \quad \rho_p = (1 - \phi_p) \rho + \phi_p \rho_a, \quad \bar{c}_p = (1 - \phi_p) \bar{c} + \phi_p c_a, \quad (\text{A.3})$$

where the density  $\rho_a$  and specific heat capacity  $c_a$  of the surrounded inert gas is similarly modeled as polynomials fitted by experimental data.

**Table A.1**  
The fitted polynomial functions of thermal properties [11,36–39].

Symbol	Definition (Unit)	Fitted polynomials
$\bar{\kappa}$	Thermal conductivity of AlSi10Mg (W/mK)	$8.36 \times 10^{-8}u^3 - 1.67 \times 10^{-4}u^2 + 0.12u + 144.76,$ $u \leq u_s$
		$-1.91u + 1257.7,$ $u_s < u < u_l$
		$0.0088u + 78.46,$ $u \geq u_l$
$\rho$	Density of AlSi10Mg (kg/m <sup>3</sup> )	$-8.63 \times 10^{-4}u^2 + 0.17u + 2634.2,$ $u \leq u_s$
		$-0.85u + 2922.6,$ $u_s < u < u_l$
		$-0.33u + 2613.2,$ $u \geq u_l$
$\bar{c}$	Specific heat capacity of AlSi10Mg (J/kgK)	$0.0011u^2 + 0.0168u + 741.16,$ $u \leq u_s$
		$0.32u + 924.04,$ $u_s < u < u_l$
		$0.0068u + 1134.3,$ $u \geq u_l$
$\rho_a$	Density of argon (kg/m <sup>3</sup> )	$3.54 \times 10^{-13}u^4 - 1.96 \times 10^{-9}u^3 + 3.93 \times 10^{-6}u^2 - 0.0036u + 1.65$
$c_a$	Specific heat capacity of argon (J/kgK)	520

**Table B.1**  
The formulas of matrices in FEM.

Symbol	Integral formula ( $i, j = 1, \dots, d, \bar{j} = 1, \dots, \bar{d}$ )	Gaussian quadrature approximation
$K_n$	$K_{n_{ij}} = \int_{\Omega} \bar{\kappa}(u_n) \nabla \phi_i \cdot \nabla \phi_j dx$	$W_k^T D_k W_k$
$M_n$	$M_{n_{ij}} = \int_{\Omega} \rho(u_n) c(u_n) \phi_i \phi_j dx$	$\sum_{\tau=1}^{n_m} W_m^{(\tau)T} D_m^{(\tau)} W_m^{(\tau)}$
$R_n$	$R_{n_{ij}} = \int_{\Gamma_r \cup \Gamma_c} \sigma \varepsilon u_n^3 \phi_i \phi_j ds$	$\sum_{\tau=1}^{n_r} W_r^{(\tau)T} D_r^{(\tau)} W_r^{(\tau)}$
$C$	$C_{ij} = \int_{\Gamma_c \cup \Gamma_s} h \phi_i \phi_j ds$	$\sum_{\tau=1}^{n_c} W_c^{(\tau)T} D_c^{(\tau)} W_c^{(\tau)}$
$a$	$a_i = \int_{\Gamma_c \cup \Gamma_s} (hu_a + \sigma \varepsilon u_a^4) \phi_i ds$	$W_a^T d_a$
$l$	$l_i = \int_{\Gamma_l \cup \Gamma_b} q_n \phi_i ds$	$\sum_{\tau=1}^{n_l} W_l^{(\tau)T} d_l^{(\tau)}$
$\bar{K}_n$	$\bar{K}_{n_{ij}} = \int_{\Omega} \bar{\kappa}(u_n) \nabla \phi_i \cdot \nabla \phi_j dx$	$W_k^T D_k \bar{W}_k$
$\bar{R}_n$	$\bar{R}_{n_{ij}} = \int_{\Gamma_r \cup \Gamma_c} \sigma \varepsilon u_n^3 \phi_i \phi_j ds$	$\sum_{\tau=1}^{n_r} W_r^{(\tau)T} D_r^{(\tau)} \bar{W}_r^{(\tau)}$
$\bar{C}$	$\bar{C}_{ij} = \int_{\Gamma_c \cup \Gamma_s} h \phi_i \phi_j ds$	$\sum_{\tau=1}^{n_c} W_c^{(\tau)T} D_c^{(\tau)} \bar{W}_c^{(\tau)}$

## Appendix B. The details in FEM

### B.1. The details of $A_n(u_n)$ and $b_n(u_n)$ in Eq. (7)

$A_n(u_n) \in \mathbb{R}^{d \times d}$  and vector  $b_n(u_n) \in \mathbb{R}^d$  for  $n = 1, 2, \dots$  are given by

$$A_n(u_n) = K_n(u_n) + \frac{1}{\Delta t} M_n(u_n) + R_n(u_n) + C, \quad (B.1)$$

$$b_n(u_n) = \frac{1}{\Delta t} M_n(u_n) u_{n-1} + l_n + a - (\bar{K}_n(u_n) + \bar{R}_n(u_n) + \bar{C}) \bar{u}_n, \quad (B.2)$$

where the matrices and vectors at the time instant  $n\Delta t$  and  $(n-1)\Delta t$  respectively have the subscript  $n$  and  $n-1$ . On the left hand side of Eq. (7), we have matrices defined on DoF including the stiffness matrix  $K_n \in \mathbb{R}^{d \times d}$ , the mass matrix  $M_n \in \mathbb{R}^{d \times d}$ , the nonlinear part of radiation heat loss  $R_n \in \mathbb{R}^{d \times d}$ , and the convection heat loss  $C \in \mathbb{R}^{d \times d}$ . On the right hand side, however, we have terms defined on the DoF and the terms related to  $\bar{d}$  nodes on the boundary  $\Gamma_b$  which are subtracted to balance the  $d$ -dimensional left hand side. In specific, the mass matrix  $M_n$ , the previous temperature  $u_{n-1} \in \mathbb{R}^d$ , the load vector  $l_n \in \mathbb{R}^d$ , and the constant part of heat loss  $a \in \mathbb{R}^d$  are defined on DoF.  $\bar{K}_n \in \mathbb{R}^{d \times \bar{d}}$ ,  $\bar{R}_n \in \mathbb{R}^{d \times \bar{d}}$  and  $\bar{C} \in \mathbb{R}^{d \times \bar{d}}$  are the coefficients of  $\bar{d}$  constant temperature nodes  $\bar{u}_n = u_b \mathbf{1} \in \mathbb{R}^{\bar{d}}$ .

### B.2. The formulas of matrices in FEM

The matrices in FEM are defined as the integrals over the domain or the boundary, the integrands of which are specified in Table B.1 where  $\phi_i$  is the basis function of node  $i$  in FEM. The Gaussian quadrature approximation of these integrals are also listed in Table B.1.  $n_m$ ,  $n_r$ , and  $n_c$  respectively represent the number of integration points sampled in each element to approximate the integrals in  $M_n$ ,  $R_n$ , and  $C$  by Gaussian quadrature rules.

### B.3. The formulas of matrices in Gaussian quadrature approximation

The details of matrices in Gaussian quadrature approximation are specified in Table B.2 where  $n_k$  is the number of integration points sampled in each element to approximate the integrals in  $K_n$  by Gaussian

quadrature rules.  $|V_p|$  is the volume of the  $p$ th tetrahedron element.  $|S_q|$  is the area of the  $q$ th triangle surface. The spatial coordinates and weights for the  $\tau$ th integration points of  $K$ ,  $M$ ,  $R$ ,  $C$ ,  $a$  and  $l$  are respectively denoted as  $\{\mathbf{x}_k^{(\tau)} \in \mathbb{R}^{e \times 3}, w_k^{(\tau)} \in \mathbb{R}^e\}_{\tau=1}^{n_k}$ ,  $\{\mathbf{x}_m^{(\tau)} \in \mathbb{R}^{e \times 3}, w_m^{(\tau)} \in \mathbb{R}^e\}_{\tau=1}^{n_m}$ ,  $\{\mathbf{x}_r^{(\tau)} \in \mathbb{R}^{m \times 3}, w_r^{(\tau)} \in \mathbb{R}^m\}_{\tau=1}^{n_r}$ ,  $\{\mathbf{x}_c^{(\tau)} \in \mathbb{R}^{m \times 3}, w_c^{(\tau)} \in \mathbb{R}^m\}_{\tau=1}^{n_c}$ ,  $\{\mathbf{x}_a \in \mathbb{R}^{m \times 3}, w_a \in \mathbb{R}^m\}$ , and  $\{\mathbf{x}_l^{(\tau)} \in \mathbb{R}^{m \times 3}, w_l^{(\tau)} \in \mathbb{R}^m\}_{\tau=1}^{n_l}$ .

### B.4. The algorithm for Picard iteration

As stated in Algorithm B.1, the nonlinear heat equation is solved via Picard iterations which takes several iterations of solving  $d$ -dimensional linear equations to converge to the solution of the original nonlinear equation with error  $\|A_n u_n - b_n\|$  below a small tolerance, e.g.  $10^{-5}$  [14].

**Algorithm B.1** The Picard iteration of solving a nonlinear heat equation

**Input:** the temperature  $u_{n-1} \in \mathbb{R}^d$ , a error tolerance  $\epsilon_p \in \mathbb{R}$ , the temperature-dependent matrix  $A_n(u_n) \in \mathbb{R}^{d \times d}$  and vector  $b_n(u_n) \in \mathbb{R}^d$ .

**Output:** the temperature  $u_n \in \mathbb{R}^d$ .

- 1:  $k = 0$ .
- 2:  $u_n^{(0)} = u_{n-1}$ .
- 3: **do**
- 4:  $u_n^{(k+1)} = A_n(u_n^{(k)})^{-1} b_n(u_n^{(k)})$ .
- 5:  $e_p = \|A_n(u_n^{(k+1)}) u_n^{(k+1)} - b_n(u_n^{(k+1)})\|$ .
- 6:  $k = k + 1$ .
- 7: **while**  $e_p > \epsilon_p$
- 8:  $u_n = u_n^{(k)}$ .

## Appendix C. The computation of the relative distance matrix

To get the relative distance matrix  $Q \in \mathbb{R}^{N \times N}$ , we first construct a weighted  $g$ -neighborhood graph with  $N$  vertices representing the  $N$

**Table B.2**  
The formulas of matrices in Gaussian quadrature approximation.

Symbol	Formula ( $j = 1, \dots, d, \bar{j} = 1, \dots, \bar{d},$ $p = 1, \dots, e, q = 1, \dots, m$ )	Symbol	Formula ( $j = 1, \dots, d, \bar{j} = 1, \dots, \bar{d}$ )
$D_k$	$\begin{bmatrix} D_{k3p,3p} \\ D_{k3p-1,3p-1} \\ D_{k3p-2,3p-2} \end{bmatrix} = \left  V_p \right  \sum_{\tau=1}^{n_k} w_k^{(\tau)} \kappa(\mathbf{x}_{k_p}^{(\tau)}) \mathbf{1}$	$W_m^{(\tau)}$	$W_{m,j}^{(\tau)} = \phi_j(\mathbf{x}_m^{(\tau)})$
$D_m^{(\tau)}$	$D_{m_p}^{(\tau)} = \left  V_p \right  w_m^{(\tau)} \rho(\mathbf{x}_{m_p}^{(\tau)}) c(\mathbf{x}_{m_p}^{(\tau)})$	$W_r^{(\tau)}$	$W_{r,j}^{(\tau)} = \phi_j(\mathbf{x}_r^{(\tau)})$
$D_r^{(\tau)}$	$D_{r_{qq}}^{(\tau)} = \left  S_q \right  \sigma_r \varepsilon w_r^{(\tau)} u(\mathbf{x}_{r_{qq}}^{(\tau)})^3$	$\bar{W}_r^{(\tau)}$	$W_{r,j}^{(\tau)} = \phi_j(\mathbf{x}_r^{(\tau)})$
$D_c^{(\tau)}$	$D_{c_{qq}}^{(\tau)} = \left  S_q \right  w_c^{(\tau)} h$	$W_c^{(\tau)}$	$W_{c,j}^{(\tau)} = \phi_j(\mathbf{x}_c^{(\tau)})$
$d_a$	$d_{a_q} = \left  S_q \right  w_a (h u_a + \sigma_r \varepsilon u_a^3)$	$\bar{W}_c^{(\tau)}$	$W_{c,j}^{(\tau)} = \phi_j(\mathbf{x}_c^{(\tau)})$
$d_l^{(\tau)}$	$d_{l_q}^{(\tau)} = \left  S_q \right  w_l^{(\tau)} q_h(\mathbf{x}_{l_q}^{(\tau)})$	$W_a$	$W_{a,j} = \phi_j(\mathbf{x}_a)$
$W_k$	$\begin{bmatrix} W_{k(3p-2,j)} \\ W_{k(3p-1,j)} \\ W_{k(3p,j)} \end{bmatrix} = \nabla \phi_j$	$W_l^{(\tau)}$	$W_{l,j}^{(\tau)} = \phi_j(\mathbf{x}_l^{(\tau)})$
$\bar{W}_k$	$\begin{bmatrix} \bar{W}_{k(3p-2,j)} \\ \bar{W}_{k(3p-1,j)} \\ \bar{W}_{k(3p,j)} \end{bmatrix} = \nabla \phi_j$		

**Table D.1**  
The comparison between the thermal model with FEM in this paper and experiment results in [11] where laser power 150 W and scan speed 200 mm/s.

Type of results	Thermal model with FEM	Results in [11]
Highest temperature (°C)	910.66	887
Max temperature gradient (°C/μm)	9.41	10.6
Melt pool size (μm)	65.66 × 54.86 × 32.00	64.3 × 55.8 × 33.7

**Table D.2**  
The comparison between the thermal model with FEM in this paper and experiment results in [11] where laser power 200 W and scan speed 400 mm/s.

Type of results	Thermal model with FEM	Results in [11]
Highest temperature (°C)	1060.85	1032
Max temperature gradient (°C/μm)	11.46	13.5
Melt pool size (μm)	75.12 × 58.83 × 37.20	73.7 × 48.3 × 35.2

training temperature profiles and with edges weighted as the adjacency matrix  $G \in \mathbb{R}^{N \times N}$  which is given by

$$G_{ij} = \begin{cases} \|U_{*i} - U_{*j}\| & \text{if } U_{*j} \text{ is a neighbor of } U_{*i} \\ \infty & \text{otherwise,} \end{cases} \quad \text{for } i, j = 1, \dots, N, \quad (C.1)$$

where  $U_{*j}$  is a neighbor of  $U_{*i}$  if their Euclidean distance belongs to the  $g$  smallest values in  $\|U_{*i} - U_{*j}\|$  for  $j = 1, \dots, N$ . The pair-wise distance matrix  $Q$  is computed as the shortest path matrix of this graph via the Floyd–Warshall algorithm [40].

#### Appendix D. The comparison between the thermal model with FEM and published experiment results

See Tables D.1 and D.2.

#### References

[1] M.B.A. Tamez, I. Taha, A review of additive manufacturing technologies and markets for thermosetting resins and their potential for carbon fiber integration, *Addit. Manuf.* (2020) 101748.  
 [2] J.C. Najmon, S. Raesi, A. Tovar, Review of additive manufacturing technologies and applications in the aerospace industry, in: *Additive Manufacturing for the Aerospace Industry*, Elsevier, 2019, pp. 7–31.  
 [3] S.G. Sarvankar, S.N. Yewale, Additive manufacturing in automobile industry, *Int. J. Res. Aeronaut. Mech. Eng.* 7 (2019) 1–10.  
 [4] C. Culmone, G. Smit, P. Breedveld, Additive manufacturing of medical instruments: A state-of-the-art review, *Addit. Manuf.* 27 (2019) 461–473.

[5] D. Gunasegaram, A. Murphy, A. Barnard, T. DebRoy, M. Matthews, L. Ladani, D. Gu, Towards developing multiscale-multiphysics models and their surrogates for digital twins of metal additive manufacturing, *Addit. Manuf.* 46 (2021) 102089.  
 [6] J. Smith, W. Xiong, W. Yan, S. Lin, P. Cheng, O.L. Kafka, G.J. Wagner, J. Cao, W.K. Liu, Linking process, structure, property, and performance for metal-based additive manufacturing: Computational approaches with experimental support, *Comput. Mech.* 57 (4) (2016) 583–610.  
 [7] Z. Tong, X. Ren, J. Jiao, W. Zhou, Y. Ren, Y. Ye, E.A. Larson, J. Gu, Laser additive manufacturing of FeCrCoMnNi high-entropy alloy: Effect of heat treatment on microstructure, residual stress and mechanical property, *J. Alloys Compd.* 785 (2019) 1144–1159.  
 [8] T.P. Moran, D.H. Warner, N. Phan, Scan-by-scan part-scale thermal modelling for defect prediction in metal additive manufacturing, *Addit. Manuf.* (2020) 101667.  
 [9] M. Roy, O. Wodo, Data-driven modeling of thermal history in additive manufacturing, *Addit. Manuf.* 32 (2020) 101017.  
 [10] K.-H. Lee, G.J. Yun, Design optimization of thermally conductive support structure for laser powder-bed fusion process with part-scale thermal history, *Addit. Manuf.* (2022) 102627.  
 [11] Y. Li, D. Gu, Parametric analysis of thermal behavior during selective laser melting additive manufacturing of aluminum alloy powder, *Mater. Des.* 63 (2014) 856–867.  
 [12] P. Foteinopoulos, A. Papacharalampopoulos, P. Stavropoulos, On thermal modeling of additive manufacturing processes, *CIRP J. Manuf. Sci. Technol.* 20 (2018) 66–83.  
 [13] S. Kollmannsberger, M. Carraturo, A. Reali, F. Auricchio, Accurate prediction of melt pool shapes in laser powder bed fusion by the non-linear temperature equation including phase changes, *Integr. Mater. Manuf. Innov.* 8 (2) (2019) 167–177.  
 [14] M.G. Larson, F. Bengzon, The finite element method: Theory, implementation, and practice, *Texts Comput. Sci. Eng.* 10 (2010).  
 [15] M. Mozaffar, A. Paul, R. Al-Bahrani, S. Wolff, A. Choudhary, A. Agrawal, K. Ehmam, J. Cao, Data-driven prediction of the high-dimensional thermal history in directed energy deposition processes via recurrent neural networks, *Manuf. Lett.* 18 (2018) 35–39.

- [16] S. Zhang, A.L. Gain, J.A. Norato, Adaptive mesh refinement for topology optimization with discrete geometric components, *Comput. Methods Appl. Mech. Engrg.* 364 (2020) 112930.
- [17] N. Patil, D. Pal, B. Stucker, A new finite element solver using numerical eigen modes for fast simulation of additive manufacturing processes, in: 2013 International Solid Freeform Fabrication Symposium, University of Texas at Austin, 2013.
- [18] H. Jensen, C. Papadimitriou, Model reduction techniques for structural dynamic analyses, in: *Sub-Structure Coupling for Dynamic Analysis*, Springer, 2019, pp. 3–31.
- [19] T.P. Moran, P. Li, D.H. Warner, N. Phan, Utility of superposition-based finite element approach for part-scale thermal simulation in additive manufacturing, *Addit. Manuf.* 21 (2018) 215–219.
- [20] S. Nikolopoulos, I. Kalogeris, V. Papadopoulos, Non-intrusive surrogate modeling for parametrized time-dependent PDEs using convolutional autoencoders, 2021, arXiv preprint arXiv:2101.05555.
- [21] A.C. Rodriguez, S. Gugercin, J. Borggaard, Interpolatory model reduction of parameterized bilinear dynamical systems, *Adv. Comput. Math.* 44 (6) (2018) 1887–1916.
- [22] P. Benner, P. Goyal, Balanced truncation model order reduction for quadratic-bilinear control systems, 2017, arXiv preprint arXiv:1705.00160.
- [23] B. Kramer, K.E. Willcox, Nonlinear model order reduction via lifting transformations and proper orthogonal decomposition, *AIAA J.* 57 (6) (2019) 2297–2307.
- [24] S. Chaturantabut, D.C. Sorensen, Nonlinear model reduction via discrete empirical interpolation, *SIAM J. Sci. Comput.* 32 (5) (2010) 2737–2764.
- [25] N.V. Queipo, R.T. Haftka, W. Shyy, T. Goel, R. Vaidyanathan, P.K. Tucker, Surrogate-based analysis and optimization, *Prog. Aerosp. Sci.* 41 (1) (2005) 1–28.
- [26] M. Sheikhi, F.M. Ghaini, H. Assadi, Prediction of solidification cracking in pulsed laser welding of 2024 aluminum alloy, *Acta Mater.* 82 (2015) 491–502.
- [27] K.G. Ramanathan, S.H. Yen, High-temperature emissivities of copper, aluminum, and silver, *J. Opt. Soc. Amer. A* 67 (1) (1977) 32–38.
- [28] L.M.M. van den Bos, B. Sande, W. Bierbooms, Adaptive sampling-based quadrature rules for efficient Bayesian prediction, *J. Comput. Phys.* 417 (2020) 109537.
- [29] P. Keast, Moderate-degree tetrahedral quadrature formulas, *Comput. Methods Appl. Mech. Engrg.* 55 (3) (1986) 339–348.
- [30] J.B. Tenenbaum, V. De Silva, J.C. Langford, A global geometric framework for nonlinear dimensionality reduction, *Science* 290 (5500) (2000) 2319–2323.
- [31] W. Xing, A.A. Shah, P.B. Nair, Reduced dimensional Gaussian process emulators of parametrized partial differential equations based on isomap, *Proc. R. Soc. Lond. Ser. A Math. Phys. Eng. Sci.* 471 (2174) (2015) 20140697.
- [32] R. Lung, Y. Wu, D. Kamilis, N. Polydorides, A sketched finite element method for elliptic models, *Comput. Methods Appl. Mech. Engrg.* 364 (2020) 112933.
- [33] L. Chen, H. Li, S. Liu, S. Shen, T. Zhang, Y. Huang, G. Zhang, Y. Zhang, B. He, C. Yang, Simulation of surface deformation control during selective laser melting of AlSi10Mg powder using an external magnetic field, *AIP Adv.* 9 (4) (2019) 045012.
- [34] C. Butler, S. Babu, R. Lundy, R. Meehan, J. Punch, N. Jeffers, Effects of processing parameters and heat treatment on thermal conductivity of additively manufactured AlSi10Mg by selective laser melting, *Mater. Charact.* 173 (2021) 110945.
- [35] S. Torquato, H.W. Haslach Jr., Random heterogeneous materials: Microstructure and macroscopic properties, *Appl. Mech. Rev.* 55 (4) (2002) B62–B63.
- [36] C. Liu, C. Li, Z. Zhang, S. Sun, M. Zeng, F. Wang, Y. Guo, J. Wang, Modeling of thermal behavior and microstructure evolution during laser cladding of AlSi10Mg alloys, *Opt. Laser Technol.* 123 (2020) 105926.
- [37] W.J. Little, Tables of Thermodynamic Properties of Argon from 100 to 3000 K, Tech. Rep., Arnold Engineering Development Center Arnold AFB TN, 1964.
- [38] S. Lucas, Measurement of Cp/Cv for Argon, Nitrogen, Carbon Dioxide and an Argon + Nitrogen Mixture, University College London, 2010.
- [39] S.H.P. Chen, S.C. Saxena, Thermal conductivity of argon in the temperature range 350 to 2500 K, *Mol. Phys.* 29 (2) (1975) 455–466.
- [40] E. Mehrbani, M.H. Kahaei, Low-rank isomap algorithm, 2021, arXiv preprint arXiv:2103.04060.