Proceedings of the 2023 ASME International Design Engineering Technical Conferences & Computers and Information in Engineering Conference
IDETC/CIE 2023
August 20-23, 2023, Boston, MA, USA

DETC2023-117055

CAPTURING LOCAL TEMPERATURE EVOLUTION DURING ADDITIVE MANUFACTURING THROUGH FOURIER NEURAL OPERATORS

Jiangce Chen
Dept. of Mechanical Engineering
Carnegie Mellon University
Pittsburgh, PA, USA
jiangcec@andrew.cmu.edu

Wenzhuo Xu
Dept. of Mechanical Engineering
Carnegie Mellon University
Pittsburgh, PA, USA
wxu2@andrew.cmu.edu

Martha Baldwin
Dept. of Mechanical Engineering
Carnegie Mellon University
Pittsburgh, PA, USA
mebaldwi@andrew.cmu.edu

Björn Nijhuis
Chair of Nonlinear Solid Mechanics
University of Twente
Enschede, the Netherlands
b.nijhuis@utwente.nl

Ton van den Boogaard
Chair of Nonlinear Solid Mechanics
University of Twente
Enschede, the Netherlands
a.h.vandenboogaard@utwente.nl

Noelia Grande Gutiérrez
Dept. of Mechanical Engineering
Carnegie Mellon University
Pittsburgh, PA, USA
ngrandeg@andrew.cmu.edu

Sneha Prabha Narra
Dept. of Mechanical Engineering
Carnegie Mellon University
Pittsburgh, PA, USA
snarra@andrew.cmu.edu

Christopher McComb*
Dept. of Mechanical Engineering
Carnegie Mellon University
Pittsburgh, PA, USA
ccm@cmu.edu

ABSTRACT

High-fidelity, data-driven models that can quickly simulate thermal behavior during additive manufacturing (AM) are crucial for improving the performance of AM technologies in multiple areas, such as part design, process planning, monitoring, and control. However, the complexities of part geometries make it challenging for current models to maintain high accuracy across a wide range of geometries. Additionally, many models report a low mean square error (MSE) across the entire domain (part). However, in each time step, most areas of the domain do not experience significant changes in temperature, except for the heat-affected zones near recent depositions. Therefore, the MSE-based fidelity measurement of the models may be overestimated.

This paper presents a data-driven model that uses Fourier Neural Operator to capture the local temperature evolution during the additive manufacturing process. In addition, the authors propose to evaluate the model using the R^2 metric, which provides a relative measure of the model's performance compared to using mean temperature as a prediction. The model was tested on numerical simulations based on the Discontinuous Galerkin Finite Element Method for the Direct Energy Deposition process, and the results demonstrate that the model achieves high fidelity as measured by R^2 and maintains generalizability to geometries that were not included in the training process.

1 Introduction

Additive manufacturing (AM) has become more than a niche laboratory technology and is becoming increasing vital across

*Address all correspondence to this author.

Copyright © 2023 by ASME

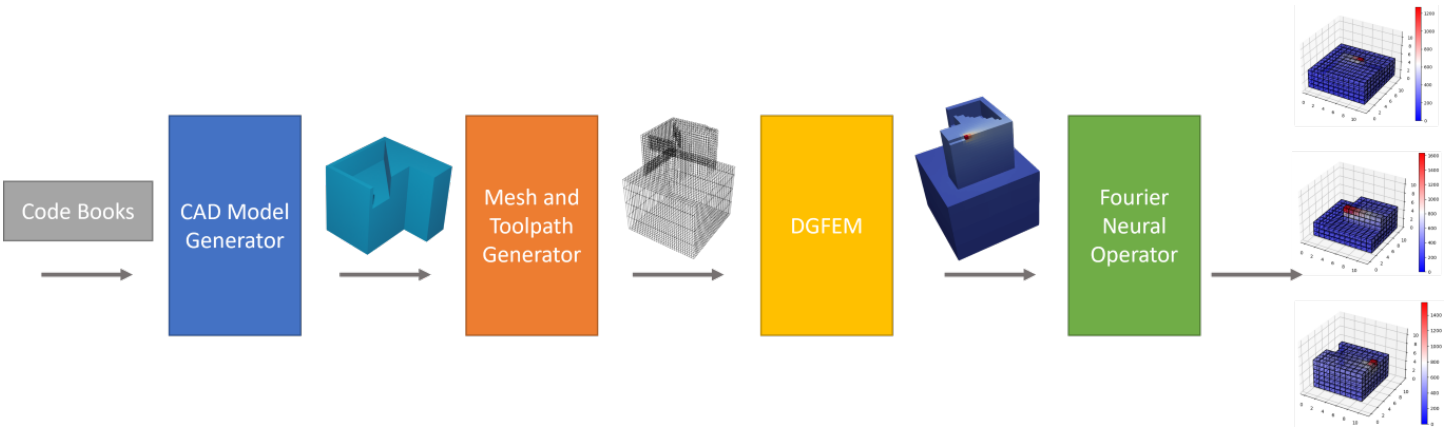


FIGURE 1. The overview of the framework proposed in this work, which trains a data-driven geometry-agnostic model to capture the local temperature evolution during AM process. An autoregressive generative model, SkexGen [1], takes disentangled codebooks to construct CAD models in the variations of encoded topological, geometric, and extrusion features. The hexahedron mesh and the toolpath are generated automatically by a code we developed. A physics-based simulation model based on Discontinuous Galerkin Finite Element Method is employed to generate the ground truths of temperature evolutions. Then, a machine learning model based on Fourier Neural Network is trained to predict the local temperature evolution in the windows around concerned regions.

various industries. This is largely due to its enormous potential in fabricating complex parts at a lower cost compared to traditional methods like machining or casting. Among the various AM technologies, Directed Energy Deposition (DED) processes have garnered considerable attention because they can build large-scale metal parts up to several meters in size at high deposition rates [2]. This paper therefore focuses on DED processes.

However, the adoption of AM is hindered by concerns regarding product quality and process efficiency [3]. For instance, Glerum et al. [4] identified inconsistencies in the AM process where the same process parameters can lead to varying material properties. Thus, establishing process-structure-property relationships for AM has become a critical research area in AM technologies [5, 6, 7, 8, 9, 10]. The temperature history resulting from the process parameters is a key determinant in several aspects, including melt pool characteristics [11], formation of defects such as lack of fusion and hot cracking [12, 13], metal grain structure [14], and residual stresses caused by high thermal gradients [15]. However, obtaining experimental data on temperature history is expensive and also challenging to capture at every point of a part. Therefore, numerical simulation is an attractive, less-expensive alternative for obtaining comprehensive temperature history data at scale. While many high-fidelity, physics-based simulation models have been developed, such as the method based on Discontinuous Galerkin Finite Element Method (DGFEM) [16], their time cost is still too high for many crucial applications. For example, iterative optimizations such as design for AM [17] and process parameter planning [18] require a massive number of thermal simulations. Additionally, real-time simulation is needed for in-situ process monitoring and control [19, 20].

Therefore, data-driven models have gained attention as a means to quickly simulate thermal behavior during AM. Mozaffar et al. [21] developed a data-driven machine learning (ML) model based on recurrent neural networks (RNN) to predict the thermal histories of arbitrary points in a part built by the DED process. For real-time temperature prediction, Paul et al. [22] proposed a framework based on extremely randomized trees (ERT), which is an ensemble of bagged decision trees that use temperatures of prior voxels and laser information as inputs to predict temperatures of subsequent voxels. Stathatos et al. [23] proposed an artificial neural network (ANN) that predicts in real-time the evolution of temperature and density for arbitrary long tracks. Roy et al. [24] observed that the AM process exhibits a high level of redundancy and periodicity and introduced a geometry representation that extracts features directly from the GCode for a ML model, such as local distances from heat sources and cooling surfaces. However, these models can only achieve good accuracy in predicting the thermal evolution of samples with geometries similar to those in the training dataset and may not perform well on more complex geometries unseen in the training process.

In order to improve the generalizability of ML models for thermal history prediction in complex geometries, Ness et al. [25] developed an ERT model that utilizes engineered features based on the underlying physics of the thermal process. Additionally, Mozaffar (2021) [3] proposed a geometry-agnostic data-driven model using (Graph Neural Network) GNNs to capture spatiotemporal dependencies of thermal responses in AM processes. However, current ML models are typically mesh-dependent, which can hinder their adoption in applications with finer meshes than those used for training. For example, mod-

Copyright © 2023 by ASME

els trained on low resolution meshes usually cannot be used to predict temperatures in the high-resolution meshes necessary for in-situ process control. Furthermore, although these models achieve low normalized mean squared error (MSE) over the built part, they may fail to accurately capture the dramatic temperature changes in the heat-affected zones (HAZ) of recent depositions. For instance, experiments in [3] have shown that the difference between the predicted temperature and ground truth in HAZ can be as large as $300^{\circ}C$. Given the close relationship between HAZ temperature and melt-pool dimensions, defect formation, and microstructure formation, accurate HAZ temperature prediction is critical. However, commonly used metrics, such as MSE, may not accurately reflect a model's performance in predicting temperatures in these critical areas, as areas far from the heat source often have stable low temperatures without significant changes. Therefore, even if an ML model predicts low accuracy at the HAZ of recent depositions, it may still score well in MSE-based fidelity measurements.

This paper proposes a data-driven model that utilizes a Fourier Neural Operator to capture the local temperature evolution during AM process. In addition to the current temperature, the model incorporates the heat source locations and local distances to the cooling surfaces to predict the temperature in the next time step. The contributions of this paper are summarized as follows:

1. A mesh-independent ML framework is established for temperature prediction in the AM process.
2. The ML model prioritizes capturing the temperature evolutions in the Heat Affected Zone (HAZ) near the recent deposition.
3. An automatic pipeline is built to generate a dataset of geometric models using an autoregressive generative model with customized and existing tools, which are then meshed and have toolpaths generated using code developed by the authors.

A high-fidelity thermal simulation model based on Discontinuous Galerkin Finite Element Method is then applied to obtain temperature histories for use in ML training and testing. The physical coefficients and parameters used in the numerical method have been calibrated with experimental data, as demonstrated in [16]. In addition, an R-squared (R^2) metric is proposed to measure the performance of ML models for temperature prediction in the AM process, as it provides a relative measure of the model's performance compared to using the mean temperature as a prediction. Results from numerical experiments demonstrate that the proposed model achieves high fidelity as measured by R^2 and maintains generalizability to geometries that were not included in the training process. The model and relevant dataset generation methods developed in this paper have the potential to be implemented in the creation of practical thermal simulation software for industrial applications.

Figure 1 provides an overview of the proposed framework. The remainder of the paper is organized as follows. Section 2 briefly introduces the concept of Fourier Neural Operator (FNO) and its application in approximating the solution of partial differential equations (PDEs). In Section 3, we detail the architecture of our ML model, the loss function and evaluation metrics, and provide a description of the heat-affected windows. Section 4 explains the data generation and preprocessing process for training the ML model and outlines the experiment settings. We present and discuss the experimental results in Section 5, and conclude in Section 6.

2 The Background of Fourier Neural Operator

A nonlinear operator is defined to be a mapping from a space of functions into another space of functions. Similar to the well-known universal approximation theorem that states neural networks can approximate any continuous function to arbitrary accuracy if no constraint is placed on the width and depth of the hidden layers [26], there is another approximation theorem states that neural networks can accurately approximate any nonlinear operator [27]. Such neural networks are called neural operators.

In the meantime, partial differential equations (PDEs) that describe physical phenomena can be viewed as nonlinear operators that map initial conditions (functions defined in Euclidean spaces) to solutions (functions in Euclidean space, with or without time). As a result, a line of research has emerged that utilizes neural operators to approximate the solutions of an entire family of PDEs [28, 29, 30]. In the context of predicting temperatures during an AM process, simulation and experimental data may be available in different resolutions. Unlike methods based on Convolutional Neural Networks (CNNs) that approximate PDE solutions in discretized Euclidean spaces, which are dependent on the mesh, neural operators can learn solutions with super-resolution. Neural operators trained on a low-resolution mesh can therefore be used to evaluate on a high-resolution mesh. They can thus readily be used to overcome discrepancies between the resolution of simulation and experimental data. Neural operators are therefore a suitable tool to model and predict the AM process.

The FNO [31] is a recently developed type of neural operator that utilizes the fast Fourier transform (FFT) to achieve $nlog(n)$ time complexity, in contrast to the quadratic complexity of other neural operators, like the neural operator proposed in [32]. In addition, FNO also features a noise-filtering mechanism brought by spectral analysis. While a brief introduction is provided here, readers are referred to [31] for a more detailed review.

Let D be a bounded open subset of $\mathbb{R}^d$, where d is the dimension of the Euclidean space. Let $\mathscr{A} = \mathscr{A}(D;\mathbb{R}^{d_a})$ and $\mathscr{U} = \mathscr{U}(D;\mathbb{R}^{d_u})$ be separable Banach spaces of functions that take values in $\mathbb{R}^{d_a}$ and $\mathbb{R}^{d_u}$, respectively. Typically, $\mathscr{A}$ represents the space of input functions (such as initial conditions), while $\mathscr{U}$ represents the space of output functions (i.e., solutions to PDEs).

Copyright © 2023 by ASME

Consider a non-linear operator $G^\dagger : \mathscr{A} \to \mathscr{U}$ that maps input functions to output functions. Suppose we have a set of N input-output function pairs $\{a_j, u_j\}_{j=1}^N$ observed from $\mathscr{A}$ and $\mathscr{U}$, respectively, where the set of a is selected as an independent and identically distributed sequence from the probability measure μ, and $u = G^\dagger(a)$. The goal is to approximate $G^\dagger$ by constructing a parametric non-linear operator $G_\theta : \mathscr{A} \to \mathscr{U}$, where $\theta \in \Theta$ denotes the set of parameters. To this end, we define a cost function $C : \mathscr{U} \times \mathscr{U} \to \mathbb{R}$ and seek to minimize the expected value of this cost function over the input function space, i.e.,

$$\min_{\theta \in \Theta} \mathbb{E}_{a\sim\mu}[C(G_\theta(a), u)]. \tag{1}$$

It is important to note that in practical applications, we often only have access to the point-wise evaluations of the functions a and u obtained from simulations or experiments. To simplify notation, in the following discussion, we will use a and u to refer to these numerical observations, where $a \in \mathbb{R}^{n\times d_a}$ and $u \in \mathbb{R}^{n\times d_u}$, with n being the number of sample points used to discretize the domain D. As a non-linear operator, G_θ is mesh-independent or super-resolution, which means that it can be used to evaluate functions at positions that are not in the sample points.

Directly parameterizing the non-linear operator $G^\dagger$ can be challenging. A potential solution is to use the idea of contracting neural networks, which approximate any continuous functions by breaking down the calculation into multiple layers and combining a linear transformation with a non-linear activation function in each layer. Similarly, a non-linear operator can be approximated using a series of iterative updates, where each update consists of a global linear operator and a local non-linear activation function.

The computational efficiency of neural operators is of great importance for their practical applications, as the complexities of global linear operators often lead to significant challenges in achieving efficient computation [31]. In recent years, the Fourier neural operator has emerged as a promising approach to address this issue. By implementing convolution as the global linear operator and leveraging the fast Fourier transform (FFT), this neural operator achieves $nlog(n)$ time complexity.

Convolution between two function $g : D \to \mathbb{R}^{d_v}$ and $f : D \to \mathbb{R}^{d_v}$ results in a new function $g * f : D \to \mathbb{R}^{d_v}$, which is defined as

$$g * f = \int_D g(x-y)f(y)dy.$$

This mathematical operation can be interpreted as a linear operator that transforms a function f by performing a global integral with another function g.

We can parameterize convolution by using a family of parametric kernel functions k_ϕ, where ϕ belongs to a given set of parameters Θ. With this parameterization, we can define a global linear operator as follows:

$$k_\phi * f(x) := \int_D k_\phi(x-y)f(y)dy, \tag{2}$$

where f and k_ϕ are functions defined over a domain D. This operator transforms a function f using a global integral with the kernel function k_ϕ.

The FNO calculation process can be summarized as follows. First, the input $a \in \mathscr{A}$ is lifted to a higher dimension using a local linear transformation $P : \mathbb{R}^{d_a} \to \mathbb{R}^{d_v}$, such that $v_0(x) = P(a(x))$. Next, a series of iterative updates is applied, generating $v_0 \mapsto v_1 ... \mapsto v_T$, where each v_h takes value in $\mathbb{R}^{d_v}$. Finally, the output $u(x) = Q(v_T(x))$ is projected back by a local linear transformation $Q : \mathbb{R}^{d_v} \to \mathbb{R}^{d_u}$. The iterative update is defined as

$$v_{t+1}(x) := \sigma(W v_t(x) + k_\phi * v_t(x)), \forall x \in D, \tag{3}$$

where $W : \mathbb{R}^{d_v} \to \mathbb{R}^{d_v}$ is a linear transformation, $\sigma : \mathbb{R} \to \mathbb{R}$ is a local non-linear activation function.

To enable efficient computation of the convolution operation in Equation 3, the Fourier transform is utilized. Specifically, the Fourier transform of a function $f : D \to \mathbb{R}^{d_v}$ is denoted as $\mathscr{F}(f)$, and its inverse is denoted as $\mathscr{F}^{-1}(f)$. By applying the convolution theorem, the Fourier transform of a convolution of two functions can be expressed as the component-wise product of their Fourier transforms. Thus, we have the following expression for the convolution operation in Fourier space:

$$k_\phi * v_t(x) = \mathscr{F}^{-1}(\mathscr{F}(k_\phi) \cdot \mathscr{F}(v_t))(x), \forall x \in D,$$

where $\cdot$ denotes component-wise multiplication.

Since only finite point-wise evaluations of the function v_t are available, the modes of v_t in Fourier space are finite. In order to filter out noise in frequency, only low frequency modes are retained for the neural operator. Let ξ_{max} denote the number of the modes left after filtering.

Moreover, rather than constructing a family of parametric functions for k_ϕ, a more convenient approach is to directly parameterize k_ϕ in Fourier space. Consequently, the parameterized linear operator is given by Equation 4.

$$k_\phi * v_t(x) = \mathscr{F}^{-1}(R \cdot \mathscr{F}(v_t))(x), \forall x \in D, \tag{4}$$

where R is a complex-valued $(\xi_{max} \times d_v \times d_v)-$tensor whose components are the parameters of the linear operator. When the domain D is discretized uniformly, the FFT algorithm can be applied to efficiently calculate Equation 4 with a complexity of $O(n \log n)$.

3 Method

The architecture of the ML model is shown in Figure 2. Note that the ML model does not operate on the whole domain at once, but on the smaller regions, called heat-affected windows, introduced in Section 3.1.

Copyright © 2023 by ASME

FIGURE 2. The architecture of the machine learning model. MLP represents the multilayer perceptron. FFT and IFFT represent the fast Fourier Transformation and its inverse. R represents the linear transformation in the complex space. GELU is the nonlinear activation function.

3.1 Heat-Affected Windows

Thermal simulations of additive manufacturing processes involve the solution of transient heat transfer partial differential equations (PDEs). Often elements are activated sequentially according to a toolpath prescribed on a pre-defined mesh. For a model meshed with equally-sized elements, the time step Δt between successive element activation is then determined by the element size and the tool's moving speed. When a new element is activated, the PDEs are solved with the temperature prior to activation and boundary conditions, such as heat influx, convection, and fixed temperature, as input. The output is the temperature after Δt. We have observed that over a short time period, the evolution of temperature is primarily confined in a small region. Thus, to predict the temperature at a specific position after Δt, we only need the information of its local neighbor region, not the whole domain. In heat transfer analysis, the thermal diffusivity α_p characterizes a material's rate of heat transfer. For instance, for steel, α_p is approximately 12 mm^2/s. This means that for $\Delta t = 0.1$ s, the area of the neighborhood affecting the temperature of a given position is about 1.2 mm^2. Therefore, we define the heat-affected neighborhood's characteristic radius as

$$r_c = \sqrt{\alpha_p \Delta t}. \tag{5}$$

We can choose a box region around a position whose size is about 10 times r_c to ensure that most of the necessary information is included in the box to predict the temperature for the position. These box regions are called heat-affected windows or windows in this paper.

In this study, the ML model is designed to operate solely on the heat-affected windows rather than the entire domain. This approach has two significant advantages. Firstly, by reducing the number of parameters, the size of the ML model can be significantly reduced, thereby requiring less training data. Secondly, this technique enhances the generalizability of the model for a wide variety of geometries, even with a limited training dataset. This is because two different geometries may share similar local features despite appearing vastly different. For example, Figure 4 shows two different shapes but their heat-affected windows around the boundary look similar as both of their boundaries consist of plane surfaces.

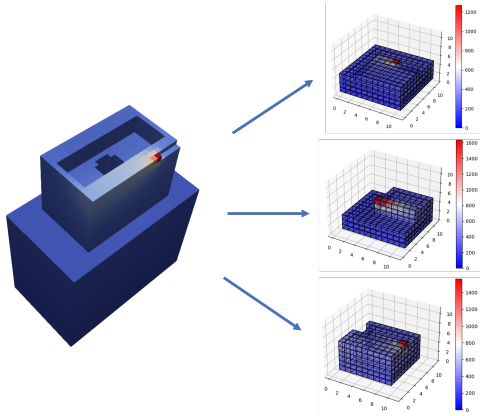


FIGURE 3. Instead of learning the temperature evolution over the domain, the ML model proposed in this paper works on the local regions cut from the domain, called heat-affected windows.

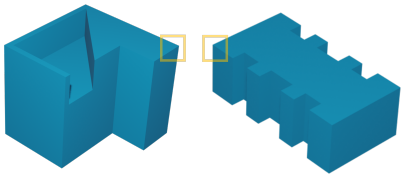


FIGURE 4. Two different shapes could look similar locally.

3.2 Input Variables

In addition to the input temperature, the ML model is also provided with other relevant information related to the heat transfer process for each element.

Activation Indicator. Heat affected windows located near the edges of the part may include void space where no material is present. To handle this case, we introduce a variable ρ_{act} that indicates whether an element is activated. Here, a value of 1 means that there is material, while a value of 0 means that there is a void.

Heat Influx Conditions. The vector $\mathbf{H}$ contains two pieces of information relevant to the heat influx condition for an element: the power of the input energy and the relative position from the center of the element to the heat influx position.

Copyright © 2023 by ASME

Boundary Impact Factors. The simulation considers two types of boundary conditions: convection-radiative and fixed-temperature conditions. The former is applied on the outer surface of the built part, while the latter is applied on the substrate bottom which the part is built on. The substrate bottom temperature is set to room temperature. To describe how each boundary condition affects local temperature evolution, boundary impact factors (BIF) **B** are defined for each element. **B** has two components corresponding to the two types of boundaries. Although there could be more complicated ways to define **B**, a simple distance-based approach is used in this work; namely, the distance between the element center and each type of boundary.

3.3 Loss Function And Evaluation Metrics

We use the normalized L_2 error as the loss function for training our model, denoted as NL_2. Let u_{pred} be the predicted temperature over a window, and u be the ground truth temperature of the window. u_{pred_i} and u_i represent the predicted and ground truth temperature of an individual element, respectively. The window is uniformly discretized with n elements. NL_2 is defined as

$$NL_2 = \sum_{i=1}^{n} \frac{\sqrt{(u_{pred_i} - u_i)^2}}{|u_i|}. \tag{6}$$

The mean squared error (MSE) is a commonly used metric for evaluating machine learning models. It is defined as the average of the squared differences between the predicted and true values, calculated as:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(u_{pred_i} - u_i)^2. \tag{7}$$

To assess the performance of ML models that make predictions on varying scales, the normalized root mean squared error (NRMSE) is often used. It is defined as:

$$NRMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(\frac{u_{pred_i} - u_i}{u_i}\right)^2}. \tag{8}$$

Here, the NRMSE takes into account the relative magnitude of the temperature values by normalizing the squared differences with respect to the ground truth temperature.

While MSE and NRMSE can reflect the overall accuracy of the ML model in many cases, they might overestimate the performance of ML models for temperature prediction of AM processes. This is because most of the domain does not experience significant temperature changes except in regions near recent deposition in a time step. Thus, a prediction that fails to capture temperature changes in these regions may still have a good MSE or NRMSE score. To address this issue, we propose to use the R^2 metric to evaluate the ML models, which measures the proportion of the variance in the ground truth that is explained by the model's predictions. It is defined as

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(u_{pred_i} - u_i)^2}{\sum_{i=1}^{n}(u_{mean} - u_i)^2}, \tag{9}$$

where u_{mean} is the average of u over the window. R^2 takes value in $(-\infty, 1]$. A negative R^2 value indicates that the model's predictions have a higher mean squared error (MSE) than a simple baseline predictor that uses the mean temperature as the prediction. When R^2 is close to zero, it suggests that the model's predictions are similar to those of the baseline predictor. Conversely, as R^2 approaches 1, the accuracy of the model's predictions improves, indicating a better fit between the model and the observed data.

4 Dataset Generation

We developed a dataset that consists of the geometric models created by a generative ML model, and a high-fidelity finite element simulation is employed to get the temperature history of all geometric points.

4.1 Geometric Model Creation

SkexGen [1] is an autoregressive generative model based on transformers that encode topological, geometric, and extrusion variations of CAD model construction sequences into disentangled code books [33]. With SkexGen, we can randomly generate geometric models with prescribed topological and extrusion features in various geometric details. In practice, we can use this CAD generative model to augment a dataset in a limited amount of time by generating synthesized models that share specific similarities with the existing dataset. Figure 5 shows the ten models created by SkexGen. Each row is generated with the same extrusion code but with different topologies and geometries. The models in the first row all have carved features in their upper region, while those in the second row are either stacked structures or contain cylindrical holes.

While it is trvial to randomly generate hundreds of geometric models, the scale of the dataset used here is bottle-necked by the intensive computation required by the thermal simulation.

The dimensions of the geometric models are normalized into a similar level before meshing. For all the models, the largest dimensions is set as 40 *mm*.

4.2 Thermal Simulation

Consider a domain Ω bounded by its boundary $\partial\Omega$, of which $\partial\Omega_{\mathrm{H}}$ represents the part at which heat is transferred to the surroundings with constant temperature T_∞, and $\partial\Omega_{\mathrm{D}}$ the part at which the temperature is fixed at T_{D}. The temperature evolution

Copyright © 2023 by ASME

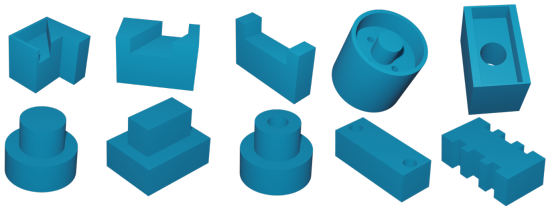


FIGURE 5. Geometric models created by SkexGen. The bounding boxes of these geometric models are in the dimensions of 40 *mm times* 40 *mm times* 40 *mm*.

within Ω is governed by the following set of PDEs:

$$\begin{aligned} \rho c_p \dot{T} &= \nabla \cdot (k_p \nabla T), && \forall \mathbf{x} \in \partial\Omega \\ -\mathbf{n} \cdot k_p \nabla T &= h_c (T - T_\infty), && \forall \mathbf{x} \in \partial\Omega_H \\ T &= T_D, && \forall \mathbf{x} \in \partial\Omega_D. \end{aligned} \tag{10}$$

Here, ρ, c_p and k_p are the temperature-dependent density, specific heat capacity and conductivity of the material, respectively. The vector $\mathbf{n}$ is the unit outward normal of the boundary at coordinate $\mathbf{x}$. In Equation 10, h_c is a temperature-dependent heat transfer coefficient that accounts for free convection with convection coefficient $h_\infty = 15W/\left(m^2K\right)$ and radiation with emissivity $\varepsilon = 0.35$ as

$$h_c(T) = \varepsilon \sigma_b \left(T^3 + T^2 T_\infty + T T_\infty^2 + T_\infty^3\right), \tag{11}$$

where σ_b is the Stefan-Boltzmann constant.

We utilized the thermal simulation algorithm developed in [16] to efficiently and accurately solve the partial differential equations described in Equation 10 for the DED process. This algorithm uses the discontinuous Galerkin finite element method (DGFEM) to spatially discretize the problem and the explicit forward Euler timestepping scheme to advance the solution in time. The algorithm activates elements based on the predefined toolpath. Newly-deposited elements are initialised at elevated temperature, after which they are allowed to cool according to Equation 10. To ensure that the high process heat input is captured correctly, newly-deposited elements are assigned an enhanced heat capacity prior to their solidification.

Our simulations utilized S355 structural steel as the material, with material properties as given in [16]. The activation temperature of newly-added elements was set to $1750°C$ and the enhanced specific heat capacity to $4537.9J/(kgK)$. The temperature of the substrate's bottom face is kept fixed at $T_\infty = 25°C$. On all other faces, convection and radiation to the surrounding air at T_∞ is modelled using Equation 11. We set the tool moving speed to $5mm/s$. All geometric models were discretized with a resolution of $20 \times 20 \times 20$, with an element size of $2mm$.

4.3 Data Preprocess and Training Setting

An efficient software for data preprocessing of thermal simulation have been developed by the authors to generate hexahedron meshes from geometric models created by SkexGen. It can add a coarse mesh for the substrate. Additionally, a toolpath that specifies the locations of the deposition tool in a sequence of time steps can be constructed based on the process parameters. The code is publicly available on GitHub [1]. An example of the mesh and toolpath generated by the code is shown in Figure 6.

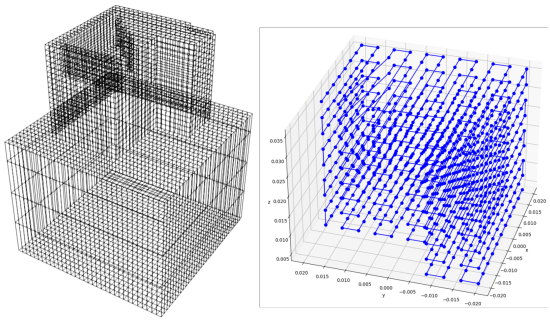


FIGURE 6. The mesh (with substrate) and the "zigzag" toolpath automatically generated by the algorithm.

The simulation generates the temperatures of all activated elements at each time step. Let $E = e_0, e_1, ..., e_m$ be the set of elements ordered by their activation sequence, where e_i is the element deposited at time step t_i. The temperature at time step t_i serves as the input to predict the temperature at time step t_{i+1}, which is calculated by the numerical simulation and serves as the ground truth for the prediction. To focus the ML model on the temperature evolution in the HAZ near the most recent deposition, heat-affected windows with dimensions $11 \times 11 \times 11$ are constructed around the deposited elements. Specifically, windows around the elements $e_{i-9}, e_{i-8}, ..., e_i$ are initially selected to train the ML model. Subsequently, more windows can be included to allow the ML model to predict the temperature of the entire domain. By doing so, we can let the ML model prioritize on the HAZ of the recent deposition. Table 1 provides the number of windows for each geometric model. To maximize the use of our limited dataset, we performed a procedure similar to k-fold cross-validation to evaluate the generalizability of the ML model to unseen geometries during training. We conducted 10 rounds of training and validation, with each round using a different geometry for validation and the remaining nine for training and testing. The windows from the nine geometries were mixed and randomly divided into two sets: 90% for training and 10% for testing. We trained the model for 50 epochs using Adam as optimizer with a learning rate of 1×10^{-3} and weight decay of 1×10^{-4}. After training, we used the ML model to predict the temperatures of the validation geometry.

5 Results and Discussion

Table 2 displays the training and test metrics after 50 training epochs of each cross-validation round. The MSE, NL_2, and

[1] https://github.com/Jiangce2017/hammer_chuizi.git

Copyright © 2023 by ASME

Index	Geometry	Number of Windows
1		9819
2		9804
3		9784
4		9513
5		9707
6		9921
7		9977
8		9588
9		9825
10		9725

TABLE 1. The number of the windows selected from the thermal simulation of each geometric model.

R^2 values are the average quantities across all windows in the training, test, or validation dataset. The high degree of consistency between the training and testing metrics indicates that the ML model is capable of accurately capturing temperature evolution without overfitting. It should be noted that the ground truth and prediction temperatures used for evaluation are their original values without normalization, and the temperature unit is Celsius. The relatively large variance of MSE is likely due to the variance of temperature responses in the AM process across different geometries.

Figure 7 shows the histories of these metrics during the training process. MSE and NL_2 decrease rapidly in all rounds, and R^2 quickly approaches 1, indicating that the optimizer converges successfully. MSE provides a straightforward way to describe the average error square between the prediction and ground truth. For example, the test MSE of No.1 round converges to about 100, indicating that the average absolute difference between prediction and ground truth is approximately $10^{\circ}C$ which is an acceptable error as the highest temperature could achieve $1500^{\circ}C$ in the simulation of AM process. NL_2 measures a comparative error that is divided by the value of the ground truth. This is why the cross-validation rounds approach similar values of NL_2 at the end. However, interpreting the ML model's performance in capturing the dramatic temperature evolution from NL_2 and MSE is challenging if most windows are far from the recent deposition and do not have significant temperature changes. R^2 reveals the relative accuracy of the prediction compared with using the mean of ground truth as the prediction. In other words, it measures the proportion of the variance in the ground truth that the prediction explains. The fact that the test R^2 in all rounds is above 0.99 suggests that the model can capture the extreme local temperature variance.

Table 3 lists the performance of the ML model on the validation geometric models, which are unseen in the training process. The ML model shows good generalizability in 7 out of 10 geometries, but fails significantly in 3 of the 10. Figure 8 visualizes the prediction results of the validation on geometric model 5. 10 windows are randomly selected for each row. At each row, the ground truth, the predicted temperature, the difference between the prediction and the temperature, and the error percent distribution over voxels of the window are shown. As we can see, the model can predict the temperature precisely for this validation geometric model, with errors ranging within 3% for most of the windows. The largest errors tend to appear near the recent deposition, with temperatures there being slightly overestimated by the ML model.

The ML model appears to fail completely at predicting the temperature for geometric models 6, 7, and 8, despite performing well in learning their local temperature evolution when included in the training process. To investigate how prediction errors are distributed over the windows, we analyzed the R^2 of the 10 windows with the worst predictions (lowest R^2) in each cross-validation round, as shown in Table 5. Among approximately 10 thousand windows, only a few poorly predicted windows can adversely affect the average prediction.

Figure 9 depicts the predictions of 10 randomly selected windows from the cross-validation rounds on geometric model 6, which show a similar level of accuracy as those in Figure 8. Therefore, despite the three failed cross-validation rounds, most of the windows have reliable predictions. These observations align with the findings reported by [34], which suggests that most points in the parts built by the AM process experience repetitive and similar temperature histories.

This also suggests that the three geometries possess unique local features that are not present in the other geometries used in training. Thus, the current dataset may be insufficient for covering the range of common geometric models used in practice. Additionally, the large errors in these cases indicate that the ML model may overfit to the training geometries if certain representative geometries are not included. To overcome these limitations, we suggest building a larger dataset that includes comprehensive geometric features of parts built by the AM process.

Copyright © 2023 by ASME

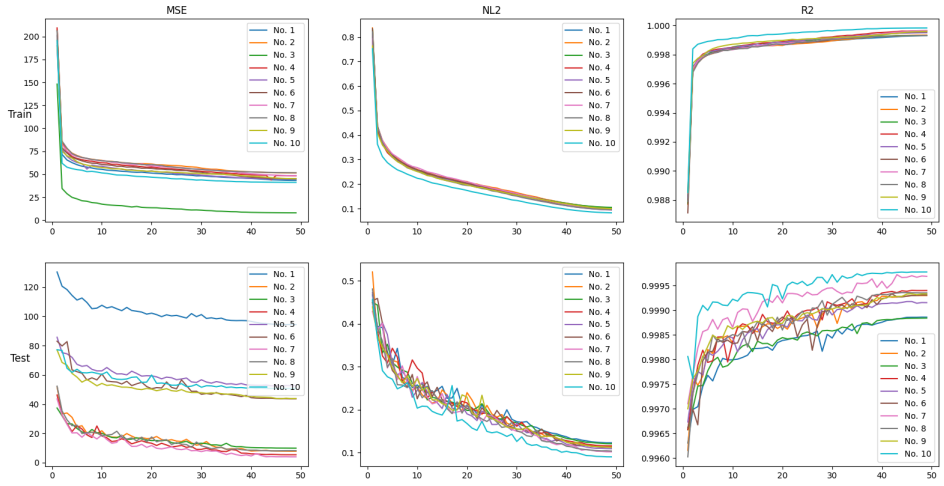
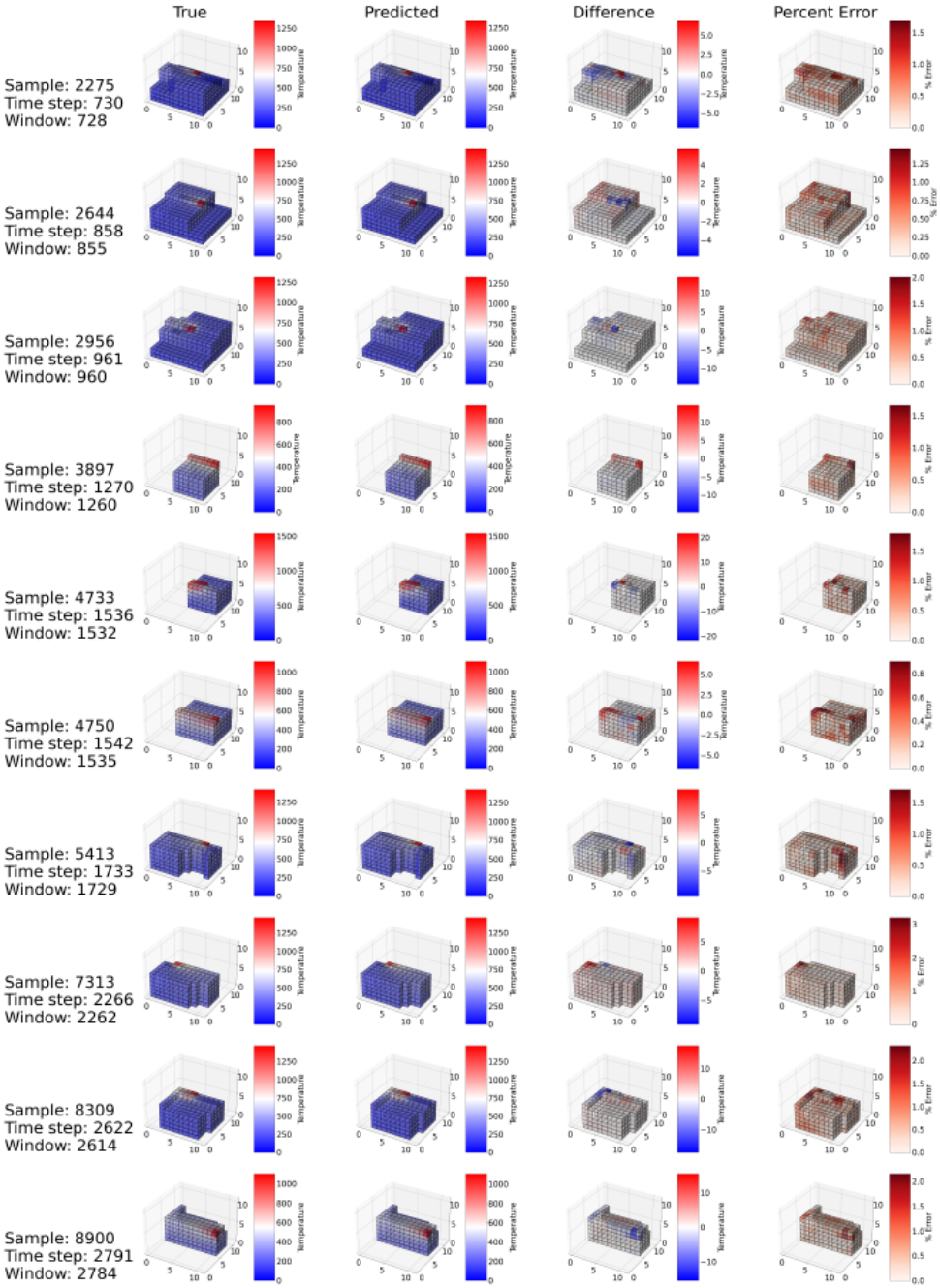


FIGURE 7. The history of the training and test evaluation metrics during the 50 epochs training.

Index	Geometry	Train MSE	Train NL_2	Train R^2	Test MSE	Test NL_2	Test R^2
1		43.2577	0.1024	0.9993	94.3523	0.1226	0.9989
2		51.4106	0.1038	0.9993	7.8131	0.1128	0.9993
3		7.9197	0.1047	0.9993	9.9346	0.1203	0.9988
4		48.4153	0.0991	0.9996	5.4243	0.1158	0.9993
5		44.5454	0.0944	0.9995	52.5766	0.1089	0.9991
6		45.1410	0.0978	0.9995	43.6450	0.1119	0.9993
7		48.4310	0.0947	0.9996	4.0064	0.1016	0.9997
8		51.5494	0.0954	0.9993	8.2268	0.1033	0.9993
9		45.1011	0.0996	0.9996	43.8328	0.1124	0.9993
10		41.1014	0.0826	0.9998	50.6812	0.0898	0.9998

TABLE 2. The training and test metrics after 50 epochs of each cross-validation round

Copyright © 2023 by ASME



Copyright © 2023 by ASME

FIGURE 8. The visualization of the prediction performance of the ML model trained validated on geometric model 5. There are 10 randomly selected windows in different timesteps. From the left to the right, each column shows the ground truth of the temperature, the prediction of the temperature, the different between the ground truth and the prediction,and the percent error, respectively. The temperature unit is Celsius.

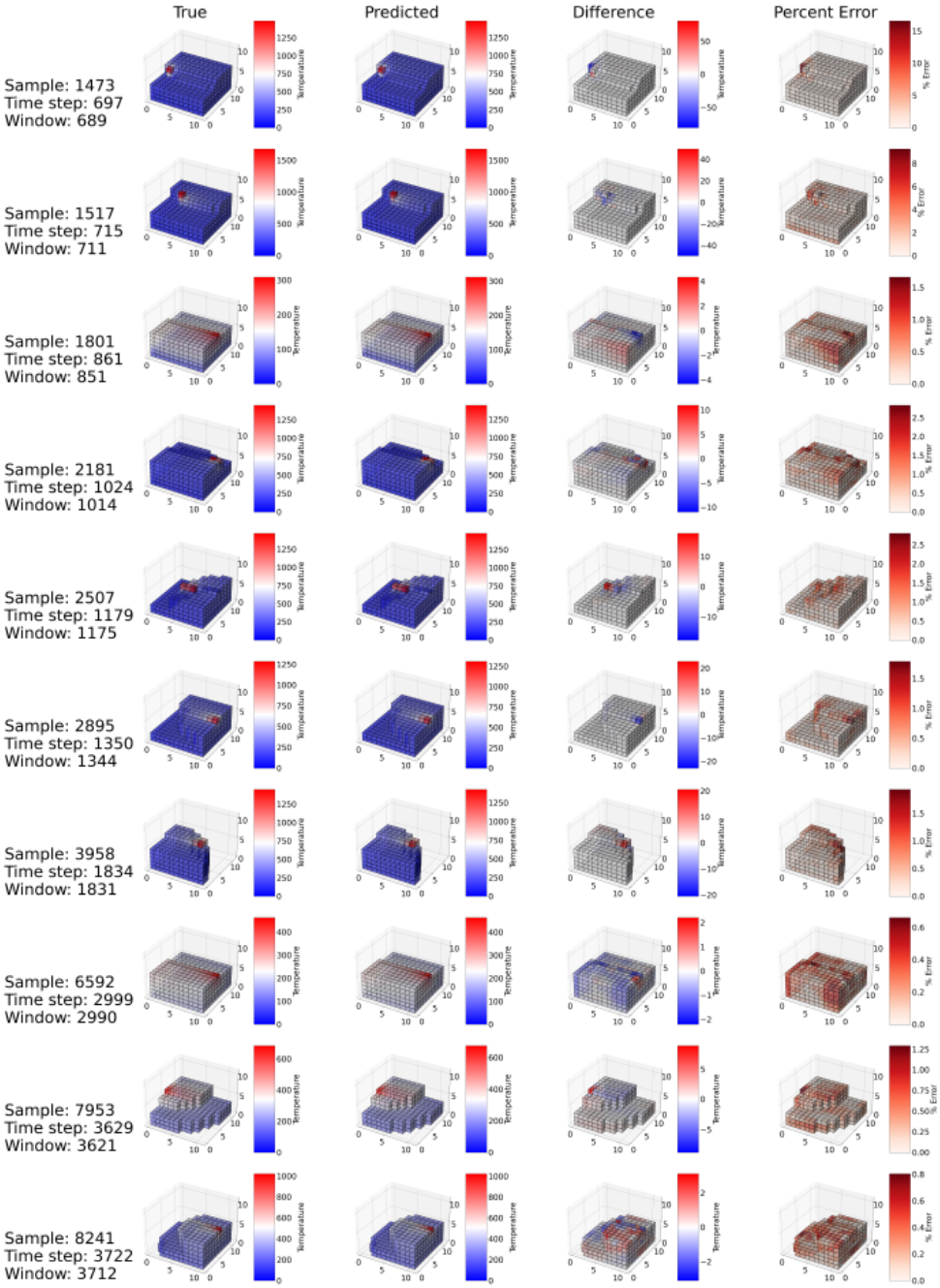


FIGURE 9. The visualization of the prediction performance of the ML model trained validated on geometric model 6.

Copyright © 2023 by ASME

Index	Geometry	*MSE*	R^2
1		7.362310	0.999586
2		0.813440	0.999953
3		131.311879	0.997899
4		2.830105	0.999730
5		1.963513	0.999787
6		4.438254×10^{11}	-2.433204×10^{7}
7		1.027052×10^{11}	-7.695437×10^{6}
8		8.096904×10^{10}	-4.151847×10^{6}
9		49.322650	0.995543
10		271.619524	0.974972

TABLE 3. The *MSE* and R^2 of each cross validation fold. The geometry shown in each row is the sample held out during training process for validation.

Index	Geometry	*MSE*	R^2
1		7.31	0.99
2		0.81	0.99
3		131.31	0.99
4		2.83	0.99
5		1.96	0.99
6		4.43×10^{11}	-2.43×10^{7}
7		1.02×10^{11}	-7.69×10^{6}
8		8.09×10^{10}	-4.15×10^{6}
9		49.32	0.99
10		271.61	0.97

TABLE 4. The *MSE* and R^2 of each cross validation fold. The geometry shown in each row is the sample held out during training process for validation.

6 Conclusion

This paper presented a data-driven model that uses Fourier Neural Operator to capture the local temperature evolution during the additive manufacturing process. To prepare the training data, we employed an automatic pipeline that uses an autoregressive generative model, SkexGen, to randomly generate a diverse set of CAD models with variations in topological, geometric, and extrusion features. The toolpath and hexahedral mesh for finite element method (FEM) analysis were then generated using a code we developed. The resulting data was used to run high-fidelity, physics-based simulations using DGFEM. The simulations produced ground truth data which was then used to train the ML model.

Our experiments demonstrate that the proposed ML model can accurately capture local temperature changes, irrespective of the geometry. The R^2 metric reveals that the model can precisely predict the large variance of temperature distributions in heat-affected zones near recent depositions. However, our experiments also highlight certain limitations. Cross-validation experiments reveal that the model may fail on geometric models that are significantly different from those used in the training dataset, indicating overfitting to the training data. This may be due, in part, to the relatively small size of the current geometric model dataset. Future work will focus on creating a larger dataset to mitigate this issue. Additionally, our method currently uses only one type of toolpath and consistent tool moving speed. Inclusion of various toolpaths, power, and tool speeds is critical to improve model performance, and will be considered in future work. The architecture of the ML model would need to be modified if more external parameters are required as the dimensions of the input change. However, the total model does not need to be re-trained from scratch. With transfer learning methods [35], the layers that connect with the input layer need to be replaced and be fine-tuned with other layers on the new data. In addition, to enhance the fidelity of the ML model and accelerate the learning process, the physics-informed neural networks [36, 37] and model order reduction methods [38, 39] could be included into our framework.

Copyright © 2023 by ASME

Index	1st	2nd	3th	4th	5th	6th	7th	8th	9th	10th
1	0.6698	0.6761	0.7180	0.8004	0.8875	0.8897	0.9064	0.9210	0.9348	0.9351
2	0.9981	0.9983	0.9987	0.9988	0.9989	0.9989	0.9989	0.9991	0.9991	0.9992
3	0.1713	0.1840	0.6392	0.6792	0.7008	0.7270	0.7440	0.7532	0.7603	0.7729
4	0.7132	0.7459	0.8119	0.9417	0.9428	0.9630	0.9722	0.9764	0.9787	0.9787
5	0.7993	0.8282	0.9115	0.9178	0.9285	0.9491	0.9501	0.9609	0.9645	0.9751
6	-8×10^{10}	-8×10^{10}	-8×10^{10}	-5×10^{10}	-4×10^{10}	-2×10^{10}	-9×10^{9}	0.3752	0.6553	0.8646
7	-4×10^{10}	-3×10^{10}	0.8019	0.9966	0.9968	0.9976	0.9979	0.9979	0.9982	0.9984
8	-4×10^{10}	0.8593	0.8671	0.8702	0.8718	0.8725	0.8868	0.8911	0.8927	0.8930
9	0.3713	0.3943	0.4016	0.4184	0.4310	0.4460	0.4508	0.4523	0.4549	0.4567
10	0.1106	0.1170	0.1281	0.1480	0.1566	0.1949	0.2295	0.3130	0.3327	0.3871

TABLE 5. The R^2 of the 10 worst window temperature predicted in the 10 cross-validation rounds.

Acknowledgements

This material is based upon work supported by the National Science Foundation through Grant No. CMMI-1825535 and by Carnegie Mellon University's Manufacturing Futures Institute through the MFI Postdoctoral Fellowship Program. Any opinions, findings, conclusions, or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of the sponsors.

REFERENCES

[1] Xiang Xu, Karl DD Willis, Joseph G Lambourne, Chin-Yi Cheng, Pradeep Kumar Jayaraman, and Yasutaka Furukawa. Skexgen: Autoregressive generation of cad construction sequences with disentangled codebooks. *arXiv preprint arXiv:2207.04632*, 2022.

[2] Thomas Lehmann, Dylan Rose, Ehsan Ranjbar, Morteza Ghasri-Khouzani, Mahdi Tavakoli, Hani Henein, Tonya Wolfe, and Ahmed Jawad Qureshi. Large-scale metal additive manufacturing: a holistic review of the state of the art and challenges. *International Materials Reviews*, 67(4):410–459, 2022.

[3] Mojtaba Mozaffar, Shuheng Liao, Hui Lin, Kornel Ehmann, and Jian Cao. Geometry-agnostic data-driven thermal modeling of additive manufacturing processes using graph neural networks. *Additive Manufacturing*, 48:102449, 2021.

[4] Jennifer Glerum, Jennifer Bennett, Kornel Ehmann, and Jian Cao. Mechanical properties of hybrid additively manufactured inconel 718 parts created via thermal control after secondary treatment processes. *Journal of Materials Processing Technology*, 291:117047, 2021.

[5] Nadia Kouraytem, Xuxiao Li, Wenda Tan, Branden Kappes, and Ashley D Spear. Modeling process–structure–property relationships in metal additive manufacturing: a review on physics-driven versus data-driven approaches. *Journal of Physics: Materials*, 4(3):032002, 2021.

[6] Qixiang Luo, Lu Yin, Timothy W Simpson, and Allison M Beese. Dataset of process-structure-property feature relationship for laser powder bed fusion additive manufactured ti-6al-4v material. *Data in Brief*, page 108911, 2023.

[7] Evdokia Popova, Theron M Rodgers, Xinyi Gong, Ahmet Cecen, Jonathan D Madison, and Surya R Kalidindi.

Copyright © 2023 by ASME

Process-structure linkages using a data science approach: application to simulated additive manufacturing data. *Integrating materials and manufacturing innovation*, 6(1):54–68, 2017.

[8] Narra S.P. Cunningham R.W. Liu H. Chen H. Suter R.M. Beuth J.L. Gordon, J.V. and A.D. Rollett. Defect structure process maps for laser powder bed fusion additive manufacturing. *Additive Manufacturing*, 36:101552, 2020.

[9] Lichao Fang, Lin Cheng, Jennifer A Glerum, Jennifer Bennett, Jian Cao, and Gregory J Wagner. Data-driven analysis of process, structure, and properties of additively manufactured inconel 718 thin walls. *npj Computational Materials*, 8(1):1–15, 2022.

[10] Aditi Thanki, Louca Goossens, Agusmian Partogi Ompusunggu, Mohamad Bayat, Abdellatif Bey-Temsamani, Brecht Van Hooreweder, Jean-Pierre Kruth, and Ann Witvrouw. Melt pool feature analysis using a high-speed coaxial monitoring system for laser powder bed fusion of ti-6al-4 v grade 23. *The International Journal of Advanced Manufacturing Technology*, 120(9):6497–6514, 2022.

[11] Peter S Cook and Anthony B Murphy. Simulation of melt pool behaviour during additive manufacturing: Underlying physics and progress. *Additive Manufacturing*, 31:100909, 2020.

[12] Lu Wang, Yanming Zhang, and Wentao Yan. Evaporation model for keyhole dynamics during additive manufacturing of metal. *Physical Review Applied*, 14(6):064039, 2020.

[13] T Mukherjee and Tarasankar DebRoy. Mitigation of lack of fusion defects in powder bed fusion additive manufacturing. *Journal of Manufacturing Processes*, 36:442–449, 2018.

[14] A Zinoviev, O Zinovieva, V Ploshikhin, V Romanova, and R Balokhonov. Evolution of grain structure during laser additive manufacturing. simulation by a cellular automata method. *Materials & Design*, 106:321–329, 2016.

[15] Wen Dong, Xuan Liang, Qian Chen, Shawn Hinnebusch, Zekai Zhou, and Albert C To. A new procedure for implementing the modified inherent strain method with improved accuracy in predicting both residual stress and deformation for laser powder bed fusion. *Additive Manufacturing*, 47:102345, 2021.

[16] Björn Nijhuis, Bert Geijselaers, and Ton van den Boogaard. Efficient thermal simulation of large-scale metal additive manufacturing using hot element addition. *Computers & Structures*, 245:106463, 2021.

[17] Mojtaba Mozaffar, Shuheng Liao, Jihoon Jeong, Tianju Xue, and Jian Cao. Differentiable simulation for material thermal response design in additive manufacturing processes. *Additive Manufacturing*, 61:103337, 2023.

[18] Zhongji Sun, Yan Ma, Dirk Ponge, Stefan Zaefferer, Eric A Jägle, Baptiste Gault, Anthony D Rollett, and Dierk Raabe. Thermodynamics-guided alloy and process design for additive manufacturing. *Nature communications*, 13(1):1–12, 2022.

[19] Dalia Mahmoud, Marcin Magolon, Jan Boer, MA Elbestawi, and Mohammad Ghayoomi Mohammadi. Applications of machine learning in process monitoring and controls of l-pbf additive manufacturing: A review. *Applied Sciences*, 11(24):11910, 2021.

[20] Youssef AbouelNour and Nikhil Gupta. In-situ monitoring of sub-surface and internal defects in additive manufacturing: A review. *Materials & Design*, page 111063, 2022.

[21] Mojtaba Mozaffar, Arindam Paul, Reda Al-Bahrani, Sarah Wolff, Alok Choudhary, Ankit Agrawal, Kornel Ehmann, and Jian Cao. Data-driven prediction of the high-dimensional thermal history in directed energy deposition processes via recurrent neural networks. *Manufacturing letters*, 18:35–39, 2018.

[22] Arindam Paul, Mojtaba Mozaffar, Zijiang Yang, Wei-keng Liao, Alok Choudhary, Jian Cao, and Ankit Agrawal. A real-time iterative machine learning approach for temperature profile prediction in additive manufacturing processes. In *2019 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, pages 541–550. IEEE, 2019.

[23] Emmanuel Stathatos and George-Christopher Vosniakos. Real-time simulation for long paths in laser-based additive manufacturing: a machine learning approach. *The International Journal of Advanced Manufacturing Technology*, 104(5):1967–1984, 2019.

[24] Mriganka Roy and Olga Wodo. Data-driven modeling of thermal history in additive manufacturing. *Additive Manufacturing*, 32:101017, 2020.

[25] Kari Lovise Ness, Arindam Paul, Li Sun, and Zhiliang Zhang. Towards a generic physics-based machine learning model for geometry invariant thermal history prediction in additive manufacturing. *Journal of Materials Processing Technology*, 302:117472, 2022.

[26] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366, 1989.

[27] Tianping Chen and Hong Chen. Universal approximation to nonlinear operators by neural networks with arbitrary activation functions and its application to dynamical systems. *IEEE Transactions on Neural Networks*, 6(4):911–917, 1995.

[28] Kaushik Bhattacharya, Bamdad Hosseini, Nikola B Kovachki, and Andrew M Stuart. Model reduction and neural networks for parametric pdes. *arXiv preprint arXiv:2005.03180*, 2020.

[29] Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Neural operator: Graph kernel network for partial differential equations. *arXiv preprint*

Copyright © 2023 by ASME

arXiv:2003.03485, 2020.

[30] Lu Lu, Pengzhan Jin, and George Em Karniadakis. Deeponet: Learning nonlinear operators for identifying differential equations based on the universal approximation theorem of operators. *arXiv preprint arXiv:1910.03193*, 2019.

[31] Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Fourier neural operator for parametric partial differential equations. *arXiv preprint arXiv:2010.08895*, 2020.

[32] Ravi G Patel, Nathaniel A Trask, Mitchell A Wood, and Eric C Cyr. A physics-informed operator regression framework for extracting data-driven continuum models. *Computer Methods in Applied Mechanics and Engineering*, 373:113500, 2021.

[33] Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017.

[34] Albert C To, Xuan Liang, and Wen Dong. Modified inherent strain method for predicting residual deformation and stress in metal additive manufacturing. *Additive Manufacturing Technology: Design, Optimization, and Modeling*, pages 219–265, 2023.

[35] Fuzhen Zhuang, Zhiyuan Qi, Keyu Duan, Dongbo Xi, Yongchun Zhu, Hengshu Zhu, Hui Xiong, and Qing He. A comprehensive survey on transfer learning. *Proceedings of the IEEE*, 109(1):43–76, 2020.

[36] Maziar Raissi, Paris Perdikaris, and George Em Karniadakis. Physics informed deep learning (part i): Data-driven solutions of nonlinear partial differential equations. *arXiv preprint arXiv:1711.10561*, 2017.

[37] Qiming Zhu, Zeliang Liu, and Jinhui Yan. Machine learning for metal additive manufacturing: predicting temperature and melt pool fluid dynamics using physics-informed neural networks. *Computational Mechanics*, 67(2):619–635, 2021.

[38] Randi Wang, Morad Behandish, and Ion Matei. Model order reduction of physical systems, February 2 2023. US Patent App. 17/386,674.

[39] Randi Wang and Morad Behandish. Surrogate modeling for physical systems with preserved properties and adjustable tradeoffs. *arXiv preprint arXiv:2202.01139*, 2022.

Copyright © 2023 by ASME