JOM, Vol. 72, No. 12, 2020
https://doi.org/10.1007/s11837-020-04438-4

TECHNICAL ARTICLE

Physics-Informed and Hybrid Machine Learning in Additive Manufacturing: Application to Fused Filament Fabrication

BERKCAN KAPUSUZOGLU [1] and SANKARAN MAHADEVAN [1,2]

1.—Department of Civil and Environmental Engineering, Vanderbilt University, Nashville, TN 37235, USA. 2.—e-mail: sankaran.mahadevan@vanderbilt.edu

This article investigates several physics-informed and hybrid machine learning strategies that incorporate physics knowledge in experimental data-driven deep-learning models for predicting the bond quality and porosity of fused filament fabrication (FFF) parts. Three types of strategies are explored to incorporate physics constraints and multi-physics FFF simulation results into a deep neural network (DNN), thus ensuring consistency with physical laws: (1) incorporate physics constraints within the loss function of the DNN, (2) use physics model outputs as additional inputs to the DNN model, and (3) pretrain a DNN model with physics model input-output and then update it with experimental data. These strategies help to enforce a physically consistent relationship between bond quality and tensile strength, thus making porosity predictions physically meaningful. Eight different combinations of the above strategies are investigated. The results show how the combination of multiple strategies produces accurate machine learning models even with limited experimental data.

INTRODUCTION

Achieving the desired material properties and product quality in additive manufacturing (AM) processes has been studied using trial-and-error experiments as well as process models [either physics-based or machine learning (ML) models]. In a trial-and-error approach, the AM process is repeated multiple times with different process parameter combinations to achieve the desired microstructure and properties of the manufactured parts; this is expensive and time-consuming. Moreover, this trial-and-error approach needs to be implemented every time a new design needs to be manufactured. Therefore, in recent years, research efforts have focused on model-based methods for optimizing the AM process parameters.

Several physics-based models have been developed depending on the AM process category and the quantity of interest (QoI).[4] Costa et al.[3] proposed an analytical solution for transient heat transfer during the printing process in fused filament fabrication (FFF). Different models have been proposed in the literature to study polymer sintering.[6,7,9,17] Many of these models are parametric representations of complex physical processes based on various approximations. The parameters of such physics-based models as well as the model errors need to be calibrated for each AM process using available observation data to reduce the uncertainty in the model predictions,[11,15,16] Due to the complex physics of the AM process, a different model is needed for each sub-stage or phenomenon in the manufacturing process to accurately predict the QoI. Furthermore, physics models with reasonable fidelity and accuracy require tremendous computational effort; as a result, the use of physics-based modeling in AM of realistic products has been challenging and limited.

Recently, several studies have used the available AM experimental data to build *black-box* ML models. In addition, with increased computing power, deep learning has become a prominent tool for solving classification and regression problems. In the context of AM, Khanzadeh et al.[13] compared supervised machine learning approaches to classify melt pools to predict porosity. An artificial neural network has been used to predict the geometry of a single bead in wire and arc additive manufacturing

(Received June 3, 2020; accepted October 5, 2020; published online October 26, 2020)

from the wire-feed rate and travel speed.[5]. Kwon et al.[14] investigated the convolutional neural network (CNN) to predict laser power from melt pool images. Zhang et al.[20] used a CNN model to perform in-process porosity monitoring of laser-based AM processes.

Physics-based models do not require large amounts of data, but are generally limited by their computational complexity or incomplete physics. In contrast, ML models appear promising for complex systems that are not fully understood or represented with simplified relationships, given adequate quality and quantity of data. However, ML models represent the complex physics without taking into account any physical laws and thus can produce results that are inconsistent with physical laws. Karpatne et al.[12] proposed the combined use of physics-based and ML models to achieve more accurate and physically consistent predictions by leveraging the advantages of each method. To make ML models consistent with physical laws, Karpatne et al. [12] incorporated physical constraints into the loss function of ML models. Another method that is also implemented by Karpatne et al.[12] to combine physics-based and ML models is to use the physics-based model outputs as additional inputs in an ML model along with other inputs. Jia et al.[10] used synthetic data generated by executing physics-based models for multiple input combinations to pre-train the ML model to leverage the knowledge embedded in physics-based models. These general ideas have been explored in multiple engineering applications, such as geoscience, fluid dynamics, and thermodynamics, and this article investigates these ideas for additive manufacturing.

The goal of this article is to enhance the experimental data-driven ML models for AM by incorporating the physics knowledge, thus helping the ML model to produce more accurate and physically meaningful results. Three different strategies are explored for this purpose: (1) incorporate physics constraints within the loss function of the deep neural network (DNN), (2) use physics model outputs as additional inputs to the DNN model, and (3) pre-train a DNN model with physics model input-output and then update it with experimental data. Eight different combinations of the above three strategies are explored, and their performance in porosity and bond quality prediction of the FFF-produced parts is examined.

The strategies investigated in this article help to better capture the physics of the AM process by leveraging physical laws while improving the generalization performance of data-driven AM models. Compared to previous studies, the proposed strategies address two different physical QoIs (neck diameter and porosity) and use multiple physics-based loss functions to enhance the ML model. The main contributions of this work are as follows:

1. Several physics-informed and hybrid machine learning models are developed for porosity prediction of FFF parts using physics constraints, a physics-based model, and experimental data, using the three strategies mentioned above.
2. An enhanced physics-based model is developed to account for realistic filament geometry and the change of geometry during the printing process.
3. The proposed models are trained and evaluated using laboratory experiments where FFF parts are printed with varying input conditions, and data are collected to measure the quality characteristics (bond quality, porosity) of the parts.

The remainder of the article is organized as follows. The proposed methodology is presented in "Methodology" section, followed by the implementation of the methodology for a FFF process in "Implementation of PIML to FFF" section. Concluding remarks are provided in "Conclusion" section.

METHODOLOGY

This section develops the proposed physics-informed machine learning (PIML) approaches to predict the bond formation and mesostructure in FFF-produced parts. The methodology is applicable to any AM process with corresponding data and physical laws for the prediction QoI. The three components of the methodology are: (1) physics-based models, (2) experiments, and (3) construction of PIML models.

Physics-Based Models

This subsection describes the two coupled multiphysics models (thermal model and polymer sintering model) that predict the bond formation between adjacent filaments and the mesostructure of the printed part. The porosity and bond quality of an FFF part is dependent on the temperature history of filaments. Thus, it is important to predict the temperature evolution of filaments to estimate the final mesostructure of the printed part. The thermal model, based on the work by Costa et al. [3], is used to predict the temperature evolution of filaments considering the material properties, part geometry, and process parameters. The output of the heat transfer model (temperature) is input to the sintering model to predict the porosity and bond quality. A new method is then developed, which considers realistic filament geometry and allows the filament geometry to change during the printing process, to compute the rate of polymer sintering and the final mesostructure of the printed part using the predicted temperature evolution of each filaments. Thus, the mapping from input to output is a multi-physics model, i.e., models of two physical phenomena (heat transfer and sintering) are combined to predict the porosity and bond quality.

FFF Temperature Modeling

Fused filament fabrication (FFF) (also known as fused deposition modeling or FDM®) is an AM technology based on material extrusion for manufacturing polymer and plastic parts. The continuous strand of material is pushed through a heated nozzle and deposited as a molten extruded thin filament onto the build plate in a predefined path to form the part in the desired shape. The schematic of the FFF process is shown in Fig. 1.

In FFF, each filament is subjected to the same heat transfer mechanism with different boundary conditions depending on thermal conditions such as the environment, build plate, and extrusion temperature as well as the part geometry, material properties, and deposition sequence. Recently, Costa et al.[3] developed an analytical solution for the transient heat transfer during the print process in FFF. The temperature prediction model considers conduction heat transfer with the build plate and adjacent filaments based on the fraction λ_i of the filament perimeter that is in contact and convection heat transfer with the environment. The details of this heat transfer model are provided in the *online supplementary material* (refer to online supplementary material).

Bond Formation Modeling

This subsection develops a new method to compute the rate of polymer sintering and the final mesostructure of the printed part. The sintering process is defined as the coalescence of particles, in which two particles of molten polymer form a homogeneous melt under the action of surface tension.[17] The sintering process for amorphous polymers is driven by the surface tension force since the mechanism is considered a Newtonian viscous flow.[18]

A Newtonian sintering model for polymers was initially developed by Frenkel[6] to predict the rate of polymer sintering. Pokluda et al.[17] developed a closed-form equation to predict the bond formation between two spherical particles based on the work balance of viscous dissipation and surface tension. Bellehumeur et al.[1] applied the model proposed by Pokluda et al.[17] to FFF for predicting the sintering between adjacent filaments as a nonlinear function of time, temperature-dependent surface tension $\Gamma(T)$ and viscosity $\eta(T)$, and an initial particle radius. The model has limitations related to the mesostructure of the filaments; specifically, the geometry of the filaments is assumed constant during the printing process. Based on the above discussion, in this article we propose a new sintering model, which considers realistic filament geometry (similar to the one proposed by Garzon et al. [7]) and also accounts for the change in the mesostructure of the filaments during the printing process.

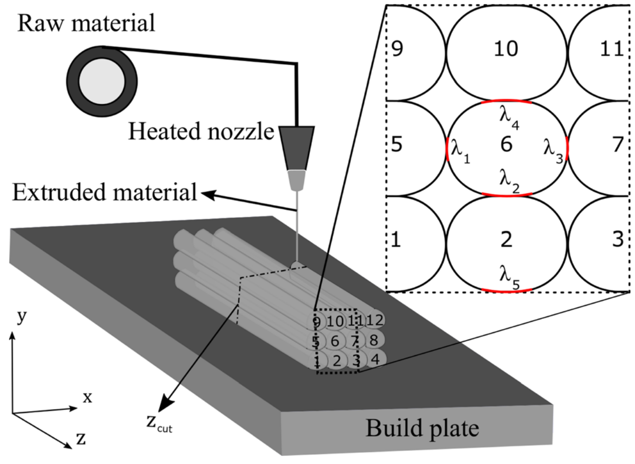


Fig. 1. Schematic of FFF.

The sintering process is simulated by considering two symmetrical adjacent filaments. In this work, all filaments are assumed to undergo the same sintering process by neglecting the effect of location. The cross-section geometry of a filament is composed of a rectangle with an initial width a_0 and two half circles with a radius of $h_0/2$ at initial time $t = 0$ as proposed by Garzon et al. [7]. At $t = 0$, adjacent filaments have one contact point between them (see Fig. 2). The width of the rectangle $a(t)$ evolves in time, together with $h(t)$ such that the width of each filament $w = a_0 + h_0 = a(t) + h(t)\cos(\theta(t))$ stays constant. The layer height $h(t)$ is assumed to evolve in time based on the experimental observations, in which the average layer height for FFF parts has decreased more than the average width of filaments (see Fig. 3). During the sintering process, the width and length of each filament (w and L) are assumed to be constant. The coalescence between these two half circles forms the sintering angle $\theta(t)$ and neck diameter $d(t)$.

To calculate the change in the filament geometry during the sintering process, the law of conservation of mass is expressed for two adjacent filaments that are assumed to have constant density:

$$2L\left(\pi\left(\frac{h_0}{2}\right)^2 + h_0 a_0\right) = 2L\left(\pi\left(\frac{h(t)}{2}\right)^2 + h(t)a(t) - 2\delta A\right), \tag{1}$$

where $2\delta A = 2(h(t)/2)^2(\theta(t) - \sin(2\theta(t)))$ is the area of intersection between two filaments as shown in Fig. 2.

Thus, the evolution of layer height is obtained as

$$h(t) = \frac{-2w + \sqrt{Q}}{\pi - 2\theta + \sin(2\theta) - 4\cos(\theta)}, \tag{2}$$

where

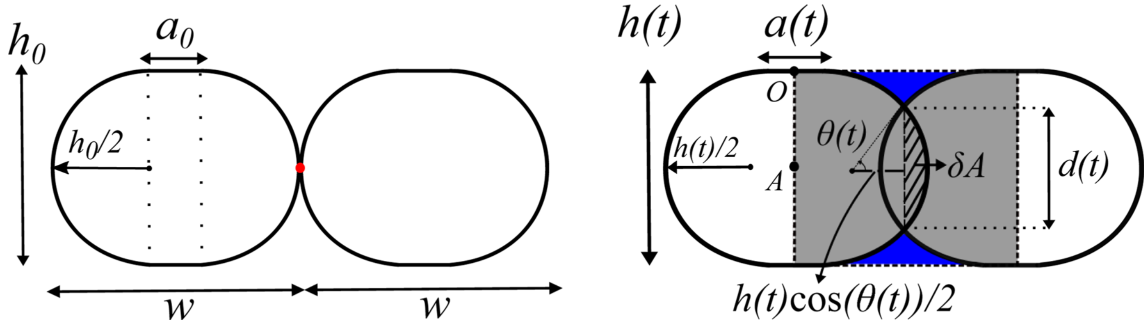


Fig. 2. Evolution of neck diameter during the sintering process.

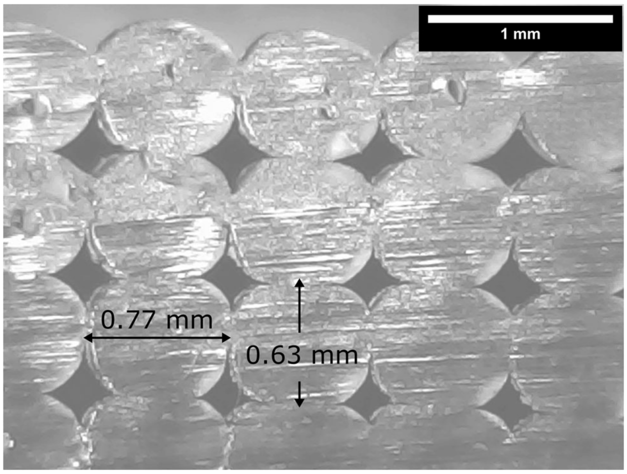


Fig. 3. Cross-sectional geometry of an FFF part printed with extrusion temperature 240°C, speed 42 mm/s, initial layer height 0.7 mm, filament width 0.8 mm, filament length 35 mm, 6 number of layers, and 15 filaments per layer.

$$Q = h_0^2\pi^2 - 2h_0^2\pi\theta + h_0^2\pi\sin(2\theta) - 4h_0^2\pi\cos(\theta) - 4h_0^2\pi + 8h_0^2\theta - 4h_0^2\sin(2\theta) + 16h_0^2\cos(\theta) + 4h_0\pi w - 8h_0\theta w + 4h_0 w\sin(2\theta) - 16h_0 w\cos(\theta) + 4w^2. \quad (3)$$

The work done by surface tension is defined as

$$W_S = -\Gamma(T)\frac{dS}{dt}, \quad (4)$$

where S is the total surface of the filaments that undergo the sintering process and is given as

$$S(t) = h(t)^4\left(-\theta + \frac{\sin(2\theta)}{2}\right) + h(t)^2\pi + 2h(t)L\pi - 4h(t)L\theta + 4h(t)a(t) + 4La(t). \quad (5)$$

Thus, applying the chain rule on Eq. 5, the work done by surface tension is obtained as

$$W_S = \Gamma(T)h(t)\left(2h(t)^3\sin^2(\theta) + 4L\right)\theta'. \quad (6)$$

The work done by the viscous forces for a Newtonian fluid can be expressed as

$$W_v = \int\int\int_V \eta(T)\nabla u : (\nabla u + \nabla u^T)dV, \quad (7)$$

V being the volume of the sintering system, and ∇u the gradient of velocity, and it is expressed as

$$\nabla u = \begin{bmatrix} \dot{\epsilon_1} & 0 & 0 \\ 0 & \dot{\epsilon_2} & 0 \\ 0 & 0 & \dot{\epsilon_3} \end{bmatrix}. \quad (8)$$

where $\dot{\epsilon_i}$ is the strain rate in ith direction.

Assuming the deformations in width and length are negligible w.r.t. deformations in height, $\dot{\epsilon_2} = \dot{\epsilon}$ is approximated by

$$\dot{\epsilon} = \frac{\partial u_x(A)}{\partial x} \approx \frac{u_x(A) - u_x(O)}{\overline{OA}_0} = \frac{\frac{d}{dt}h(t)}{h_0}. \quad (9)$$

Consequently, the work done by the viscous forces can be defined as follows

$$W_v = \int\int\int_V 2\eta(T)\dot{\epsilon}^2 dV \quad (10)$$

$$= \frac{2L\eta(T)h(t)^2(h_0\pi - 4h_0 + 4w)\sin^2(\theta)}{h_0\cos^2(\theta)}\theta''. \quad (11)$$

The evolution of the sintering angle $\theta(t)$ is then obtained by equating the work done by surface tension and the viscous forces under the assumption that θ' is always positive:[17]

$$\frac{d\theta(t)}{dt} = \frac{\Gamma(T)h_0\left(h(t)^3 + \frac{2L}{\sin^2(\theta(t))}\right)\cos^2(\theta)}{Lh(t)\eta(T)(h_0\pi - 4h_0 + 4w)}. \quad (12)$$

The evolution of neck diameter $d(t)$ with time is then computed as

$$d(t) = h(t)\sin(\theta(t)), \quad (13)$$

The porosity can be expressed using the geometry of the mesostructure shown in Fig. 2, where the shaded gray region is the filled area and the blue region is the void area. Thus, the evolution of porosity $\phi(t)$ is calculated as

$$\phi(t) = \frac{h^2 \cos(\theta) - \left(\frac{h}{2}\right)^2 \left[\pi - (2\theta - \sin(2\theta))\right]}{h^2 \cos(\theta) + ha}. \tag{14}$$

The proposed methodology is an improvement upon the previous work on sintering models by Pokluda et al.[17] and Gurrala et al.[8], because it (1) considers a realistic filament geometry based on our experiments and (2) accounts for changes in the filament geometry during the printing process. This proposed methodology is compared against experimental data in "Implementation of PIML to FFF" section and is found to give a smaller error than Gurrala et al.'s model. However, it still makes assumptions about the bonding process, and sintering is not the only physical phenomenon that takes place during the bond formation. Thus, deep learning is studied to further enhance the prediction accuracy.

PIML for Additive Manufacturing

Although physics-based models predicting the temperature evolution, bond formation, and mesostructure evolution of FFF parts are based on physical laws, they introduce bias because of incomplete representation of the complex physical process by approximating the reality. In addition, these models contain a significant number of model parameters that need to be calibrated using experimental data.[11] On the other hand, ML models are not aware of physical laws, which may result in physically inconsistent model predictions. However, they can extract complex physical relationships from available data. Thus, physics-based models and ML models can be integrated in an innovative manner to better capture the dynamics of the AM process.

In PIML models, physics knowledge and data are sought to be integrated in a synergistic manner by leveraging the complementary strengths of both models.[12] Thus, the goal is to improve the predictions beyond that of physics-based models or ML models alone by coupling physics-based models with ML models. In the following, three different strategies to combine physics knowledge and ML models are pursued: (1) incorporate physics constraints within the loss function of the DNN, (2) use physics model outputs as additional inputs to the DNN model, and (3) pre-train a DNN model with physics model input-output and then update it with experimental data.

Physics-Informed Loss Functions

A direct strategy to improve ML model predictions is by including physics-based loss functions.[12] Consider a PIML model with inputs $\boldsymbol{X}$ and outputs $\hat{\boldsymbol{Y}}$ trained using physical laws that are incorporated as constraints into the loss function:

$$\mathcal{L} = \mathcal{L}_{DNN}(\boldsymbol{Y}, \hat{\boldsymbol{Y}}) + \sum_{k=1}^{M} \lambda_{phy,k} \mathcal{L}_{phy,k}(\hat{\boldsymbol{Y}}), \tag{15}$$

where $\mathcal{L}_{DNN}$ is the regular training loss of a DNN that evaluates a supervised error [e.g., root mean squared error (RMSE); $\mathcal{L}_{DNN}(\boldsymbol{Y}, \hat{\boldsymbol{Y}}) = \sqrt{\sum_{i=1}^{n} (Y_i - \hat{Y}_i)^2 / n}$], which measures how far off the predictions $\hat{\boldsymbol{Y}}$ are from the observations $\boldsymbol{Y}$ for the n training samples, and $\mathcal{L}_{phy,k}$ is the k-th physics-based loss function, whose contribution is controlled by a hyperparameter $\lambda_{phy,k}$ and M is the total number of physics-based loss functions. The inclusion of $\mathcal{L}_{phy,k}$ ensures physically consistent model predictions [the second term of Eq. 15 means *physical inconsistency*] and can decrease the generalization error even when there is a small amount of training data.[12] In addition, $\mathcal{L}_{phy,k}$ does not require experimental observations; the data obtained from the physics model are used to evaluate physics-based loss functions.

In this work, we enforce five different physics-based loss functions (i.e., five separate physical relationships, $\mathcal{L}_{phy,k}(\hat{\boldsymbol{Y}})$, where $k = \{1, 2, 3, 4, 5\}$ and $\hat{\boldsymbol{Y}} = (\hat{Y}_1, \hat{Y}_2)$ are the overall dimensionless neck diameter (i.e., overall bond quality) and porosity predictions of FFF parts, respectively. These loss functions are defined as follows:

$$\begin{aligned}
\mathcal{L}_{phy,1}(\hat{\boldsymbol{Y}}) &= \frac{1}{N} \sum_{i=1}^{N} ReLU(-\hat{Y}_{1,i}), \\
\mathcal{L}_{phy,2}(\hat{\boldsymbol{Y}}) &= \frac{1}{N} \sum_{i=1}^{N} ReLU(\hat{Y}_{1,i} - d_{max}), \\
\mathcal{L}_{phy,3}(\hat{\boldsymbol{Y}}) &= \frac{1}{N} \sum_{i=1}^{N} ReLU(-\hat{Y}_{2,i}), \\
\mathcal{L}_{phy,4}(\hat{\boldsymbol{Y}}) &= \frac{1}{N} \sum_{i=1}^{N} ReLU(\hat{Y}_{2,i} - \phi_{0,i}), \\
\mathcal{L}_{phy,5}(\hat{\boldsymbol{Y}}) &= \frac{1}{N} \sum_{i=1}^{N} ReLU(\Delta_i),
\end{aligned} \tag{16}$$

where the first four loss functions consider the physical violations related to the overall dimensionless neck diameter and porosity across N samples and the fifth loss function represents the physical relationship between the mechanical properties and neck diameter. The physical inconsistencies in the model predictions are evaluated using these physics-based loss functions. In the first and third loss functions, negative values of neck diameter and porosity are treated as physical violations. The second loss function evaluates physically inconsistent dimensionless neck diameter predictions that are greater than the maximum dimensionless neck diameter $d_{max} = 1$. The fourth loss function penalizes the model when porosity predictions $\hat{Y}_{2,i}$ are greater than the initial porosity $\phi_{0,i}$ of the ith part. This is based on the physics knowledge that the total void area decreases as the sintering process takes place. The fifth physics-based loss function exploits the monotonic relationship between the bond quality and tensile strength of FFF-produced parts. This loss function is constructed by computing the difference in the sorted dimensionless neck diameter predictions, $\hat{Y}_{1,sorted}$, and dimensionless neck diameter predictions corresponding to sorted tensile strength estimates ($\sigma_{TS}(\hat{Y}_{2,sorted}, \xi)$), $\hat{Y}'_{1,i}$, i.e., $\Delta_i = \hat{Y}_{1,sorted,i} - \hat{Y}'_{1,i}$. The maximum stress for longitudinal raster orientation, $\sigma_{TS}(\hat{Y}_2, \xi)$ with $\xi = \{\sigma_{01}, \sigma_{02}, C_\sigma\}$ being the material parameters, is computed according to the analytical expression proposed in Garzon et al.[7] using the porosity predictions:

$$\sigma_{TS} = \sigma_{01}\left[exp\left((1-\hat{Y}_2)^{C_\sigma n_l}\right) - \hat{Y}_2\right] + \sigma_{02}(1-\hat{Y}_2), \tag{17}$$

where n_l is the number of layers of FFF parts. The tensile strength is constrained when porosity is equal to 0, i.e., $\sigma_{TS} = \sigma_{01}e + \sigma_{02}$. The overall average neck diameter and tensile strength are positively correlated, and tensile strength increases monotonically with neck diameter, whereas porosity and tensile strength are negatively correlated (as are porosity and neck diameter). More specifically, the model predictions $(\hat{Y}_{1,i}, \hat{Y}_{2,i})$ and $(\hat{Y}_{1,i+1}, \hat{Y}_{2,i+1})$ corresponding to the ith and $(i+1)$th FFF parts can be used to estimate $\sigma_{TS,i}$ and $\sigma_{TS,i+1}$. If $\sigma_{TS,i+1}$ is greater than $\sigma_{TS,i}$ — meaning the $(i+1)$th part has fewer voids than the ith part $(\hat{Y}_{2,i+1} < \hat{Y}_{2,i})$—then $\hat{Y}_{1,i+1}$ should be greater than $\hat{Y}_{1,i}$ as well by exploiting a key monotonic physical relationship between porosity and tensile strength of FFF-produced parts. Thus, with the inclusion of these physics-based penalty functions, the neck diameter and porosity predictions are ensured to be physically meaningful.

Physics Model Output as Additional ML Model Input

A physics-based model $f^{phy} : \boldsymbol{X} \rightarrow \hat{\boldsymbol{Y}}^{phy}$ can be used to predict the QoI, where $\hat{\boldsymbol{Y}}^{phy}$ are predicted estimates of the true response of the system $\boldsymbol{Y}$. A straightforward approach to combine physics-based and ML models is to use physics model output $\hat{\boldsymbol{Y}}^{phy}$ (at the experimental inputs) as additional input along with inputs $\boldsymbol{X}$, i.e., $f^{hyb} : \boldsymbol{X}^{hyb} = [\boldsymbol{X}, \hat{\boldsymbol{Y}}^{phy}] \rightarrow \hat{\boldsymbol{Y}}^{hyb}$.

Adding the physics output as an extra input to the DNN model (which is trained using experimental data) is information fusion, where the physics model is an additional source of information that is consistent with physics (i.e., when the physics model satisfies the constraints mentioned in the previous subsection). The resulting DNN model can be thought of as a hybrid model that uses the experimental data to correct the output of the physics model, which is an incomplete representation of the actual physics.

Pre-Trained PIML Model

In AM, especially in the FFF process with a not high-quality printer, parts have significant variability in quality. There is also uncertainty in measurement and lack of data due to the high cost associated with conducting experiments. Thus, data of adequate quality and quantity are important for good quality model predictions in AM.

To leverage the complex physical knowledge inherent in the physics-based models, synthetic data can be generated for multiple input combinations using physics-based models. The synthetic data can be used to train a ML model, which is used as the initial model to be updated with experimental data. The transfer of physical knowledge using a pre-trained ML model can prevent poor initialization because of lack of knowledge of the initial choice of ML model parameters prior to training. This allows the pre-trained ML model to be fine-tuned even with limited observed data. In addition, it has been shown that using synthetic data from even imperfect physics models with uncalibrated model parameters can still reduce the amount of experimental training data needed.[10]

More importantly, the pre-training can use a large amount of training data (with multiple input parameter combinations) over a wide range of values, which is not possible in experiments that could be expensive; as a result, the pre-training may help the eventual ML model to have wider generalization beyond the experimental data. This is also an important distinction of the pre-training strategy from the second strategy. Both strategies use the physics model, but in the second strategy, the physics model is only used to provide outputs corresponding to the experimental inputs, whereas in the current pre-training strategy, the physics model is used to provide outputs corresponding to a much larger set of inputs. In the numerical example in "Implementation of PIML to FFF" section, the pre-training strategy exercises the physics model over 1525 input combinations, whereas the second strategy above only employs the physics model over 39 experimental input combinations. However, the advantage of the pre-training strategy in using a larger input data set (for physics model runs) compared to the experiments becomes limited if the physics model is computationally expensive.

In this work, the ML model is pre-trained using the outputs of an uncalibrated coupled multi-physics model (i.e., neck diameter and porosity). Furthermore, the transfer of learned physical knowledge is shown to be valuable even when the input parameters of the synthetic data generated are quite different than the experimental observations. Once the ML model is pre-trained, it is fine-tuned using limited experimental observations. This helps to learn a 3D printer-specific physical process faster and with fewer samples.

The three proposed strategies to predict the QoIs are shown in Fig. 4. Figure 4a shows the first method, where the physical knowledge is included through constraints within the loss function of a DNN trained with experimental data. Figure 4b shows the second method, where the outputs of the physics model are additional inputs to the DNN model. Figure 4c shows the third method, where a DNN model is pre-trained with data generated using the physics-based model and then updated using experimental data. The proposed PIML strategies can be applied to any AM process by leveraging the physical constraints or physics-based models.

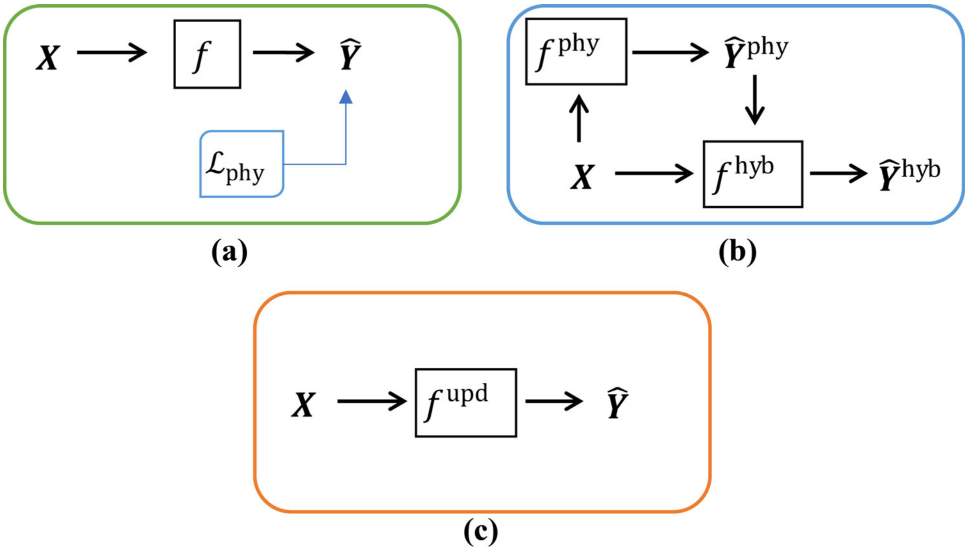
Combination of PIML Strategies

Based on the proposed three strategies to incorporate physics knowledge into the ML model, eight separate ML models can be constructed:

1. $\mathbf{DNN}$
2. $\mathbf{DNN}^{\mathcal{L}_{\mathbf{phy}}}$
3. $\mathbf{DNN}^{\mathbf{hyb}}$
4. $\mathbf{DNN}^{\mathbf{upd}}$
5. $\mathbf{DNN}^{\mathbf{hyb},\mathcal{L}_{\mathbf{phy}}}$
6. $\mathbf{DNN}^{\mathbf{upd},\mathbf{hyb}}$
7. $\mathbf{DNN}^{\mathbf{upd},\mathcal{L}_{\mathbf{phy}}}$
8. $\mathbf{DNN}^{\mathbf{upd},\mathbf{hyb},\mathcal{L}_{\mathbf{phy}}}$

In model 1, a deep neural network **DNN** is trained using only experimental data. The inputs $\boldsymbol{X}$ for this basic DNN model are the process parameters, printer extrusion temperature, speed, layer height, filament width, length, number of layers, and number of filaments per layer; the outputs are overall dimensionless neck diameter and porosity. These inputs and outputs are the same as those used in the physics-based model f^{phy}. Model 2 ($\mathbf{DNN}^{\mathcal{L}_{\mathbf{phy}}}$) pursues the first strategy: physical knowledge related to the FFF process is included through constraints within the loss function of the DNN, as shown in Eq. 15. Model 3 pursues the second strategy: a hybrid physics-based neural network $\mathbf{DNN}^{\mathbf{hyb}}$ is trained using the outputs $\hat{\boldsymbol{Y}}^{phy}$ of f^{phy} as extra inputs in addition to $\boldsymbol{X}$, i.e., $\boldsymbol{X}^{hyb} = [\boldsymbol{X}, \hat{\boldsymbol{Y}}^{phy}]$. Model 4 ($\mathbf{DNN}^{\mathbf{upd}}$) pursues the third strategy, where the weights and biases (model parameters) of all the layers *excluding* the input layer of the pre-trained network f^{pre} (which is trained with the coupled multi-physics model input-output described in "Pre-Trained PIML Model" section) are used as initial parameters for the DNN model, and these parameters are updated with experimental data.

The architecture of the pre-trained model and the updated models (Model 4, 6, 7, and 8) is the same (except the input layer which changes with different numbers of inputs). The rest of the models (5–8) represent the combinations of the three strategies. Models 5, 6, and 7 each combine any two of the three strategies, whereas model 8 combines all three strategies. Model 5 ($\mathbf{DNN}^{\mathbf{hyb},\mathcal{L}_{\mathbf{phy}}}$) combines the use of physics model outputs as additional inputs and the incorporation of physics constraints $\mathcal{L}_{phy}$ within the loss function of the DNN. Model 6 ($\mathbf{DNN}^{\mathbf{upd},\mathbf{hyb}}$)



(a) (b) (c)

Fig. 4. PIML strategies: (a) incorporate physics constraints within the loss function of the DNN, (b) use physics model outputs as additional inputs to the DNN model, and (c) pre-training a DNN model with physics model input-output and updating it with experimental data.

combines the second and third strategies, where the optimized model parameters of f^{pre} are used as the initial values and the outputs $\hat{Y}^{phy}$ of f^{phy} are included as additional inputs. $\mathbf{DNN^{upd,hyb}}$ has the same number of inputs as $\mathbf{DNN^{hyb}}$ (i.e., both include the physics model output as additional input) and uses the optimized model parameters of all the layers excluding the input layer of f^{pre} as the initial parameters before updating with experimental data. Model 7 ($\mathbf{DNN^{upd,\mathcal{L}_{phy}}}$) combines the first and third strategies. The model parameters obtained from f^{pre} are updated using the experimental data by minimizing the augmented loss function shown in Eq. 15. Model 8 ($\mathbf{DNN^{upd,hyb,\mathcal{L}_{phy}}}$) combines the use of physics model outputs as additional inputs to the updated DNN model $\mathbf{DNN^{upd}}$ (which results in $\mathbf{DNN^{upd,hyb}}$) and the physics constraints $\mathcal{L}_{phy}$ are incorporated within the loss function of $\mathbf{DNN^{upd,hyb}}$.

IMPLEMENTATION OF PIML TO FFF

In this section we demonstrate the implementation of the proposed methodology to FFF-produced parts and investigate the performance of eight PIML models described in "Combination of PIML Strategies" section. The results show that the proposed PIML models are capable of achieving physically meaningful and accurate model predictions and require fewer experiments.

Problem Setup

First, data are collected from laboratory experiments to build the prediction models for the QoI using ML techniques. The shape of the part is conceptualized, and a CAD model is visualized and then sliced in a slicing software using the defined FFF process parameters and printing path.

The print quality depends on the adhesion of the first layer with the build plate.[11] Thus, several measures are needed to ensure proper adhesion. For example, the printing environment is modified by adding an enclosure to the 3D printer to isolate the printing environment from external effects. Kapton tape is used on the glass build plate to enhance the adhesion of acrylonitrile butadiene styrene (ABS) with the build plate. After these modifications, the part is printed and then measured with appropriate monitoring techniques.

A commercial material, Ultimaker Black ABS, was extruded from an Ultimaker 2 Extended+ 3D printer to manufacture parts with unidirectionally aligned filaments. Using Latin hypercube sampling, 20 sets of process parameters are generated. The ranges considered for the variables are printer extrusion temperature T_e: (210°C - 260°C), and extrusion speed S_e: (15 mm/s - 46 mm/s). Since the values of material properties ξ do not affect the outcome of $\mathcal{L}_{phy,5}(\hat{Y})$, the values calibrated by Garzon et al.[7] are used. All specimens were sectioned at the midpoint $z_{cut} = L/2$ (since the statistical properties of QoIs along the length of the specimens were constant) to analyze the mesostructural feature of interest using microscopy images processed through the ImageJ software.[19]. The collected experimental data are subsequently used to create DNN prediction models.

Model Training and Prediction

The eight DNN models were implemented using the Keras package[2] with Tensorflow backend. The pre-trained model, f^{pre}, is first trained with physics model input-output data consisting of 1525 input parameter combinations over a range of experimental values, i.e., ($210°C \leq T_e \leq 260°C$, 15 mm/s $\leq S_e \leq$ 46 mm/s), and then updated for different combinations of the proposed strategies using observed data. (Note that in contrast only 39 physical experiments with 20 unique input parameter combinations are available; see Fig. 5). The input data of the training and test sets are normalized prior to the training of the DNN models (the output quantities are dimensionless and between 0 and 1), and the hyperparameters of these models are tuned with grid search ($\lambda_{phy} = 0.3, 0.3, 0.15, 0.15, 0.008$). Fully connected DNN models with two hidden layers and ten neurons in each hidden layer are constructed, and the weights of all neurons in these models are uniformly randomly initialized between 0 and 1. L1 and L2 regularizers are used as a penalty term to avoid overfitting. The Rectified Linear Unit (ReLU) activation function and Adam optimizer are used to perform stochastic gradient descent in learning the model parameters.

The number of epochs for the convergence of training is approximately the same for each model except f^{pre}, which converges in 40 epochs. The computation time for training of each model is on average 15 s using a desktop computer (Intel®

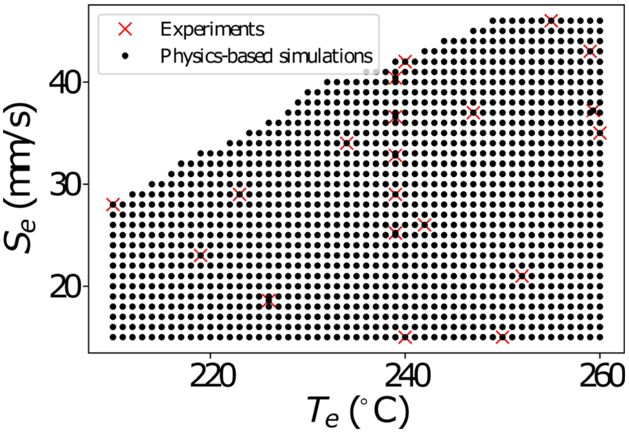


Fig. 5. Design of experiments for physics-based simulations and experiments.

Table I. Effect of different amounts of training data on the RMSE of different ML models

Model	20%	30%	40%	50%	100%	Mean physical inconsistency
f^{phy}	–	–	–	–	0.112	**0.000**
1. **DNN**	0.101(± 0.022)	0.084(± 0.018)	0.044(± 0.017)	0.028(± 0.004)	0.025(± 0.005)	0.201
2. $\mathbf{DNN}^{\mathcal{L}_{\mathbf{phy}}}$	0.055(± 0.017)	0.050(± 0.011)	0.025(± 0.005)	0.024(± 0.007)	0.021(± 0.005)	**0.000**
3. $\mathbf{DNN}^{\mathbf{hyb}}$	0.098(± 0.021)	0.078(± 0.024)	0.044(± 0.016)	0.028(± 0.007)	0.024(± 0.005)	0.195
4. $\mathbf{DNN}^{\mathbf{upd}}$	0.058(± 0.012)	0.049(± 0.011)	0.026(± 0.005)	0.025(± 0.005)	0.020(± 0.005)	0.133
5. $\mathbf{DNN}^{\mathbf{hyb},\mathcal{L}_{\mathbf{phy}}}$	0.059(± 0.018)	0.041(± 0.012)	0.026(± 0.004)	0.023(± 0.004)	0.020(± 0.005)	**0.000**
6. $\mathbf{DNN}^{\mathbf{upd},\mathbf{hyb}}$	0.057(± 0.014)	0.047(± 0.011)	0.026(± 0.003)	0.026(± 0.005)	0.024(± 0.006)	0.134
7. $\mathbf{DNN}^{\mathbf{upd},\mathcal{L}_{\mathbf{phy}}}$	0.054(± 0.011)	0.045(± 0.013)	0.026(± 0.003)	0.025(± 0.005)	0.021(± 0.005)	**0.000**
8. $\mathbf{DNN}^{\mathbf{upd},\mathbf{hyb},\mathcal{L}_{\mathbf{phy}}}$	0.053(± 0.013)	0.040(± 0.010)	0.025(± 0.004)	0.023(± 0.004)	0.018(± 0.003)	**0.000**

Xeon® CPU E5-1660 v4@3.20GHz with 32 GB RAM and GPU NVIDIA Quadro K620 with 2 GB).

Model Performance

To measure the accuracy of the trained DNN models, the model predictions of the test data are compared with the observed overall neck diameters and porosity. Each model is trained 30 times and compared against the FFF experimental data not used for training (i.e., data from 19 parts) to evaluate the mean and standard deviation of RMSE and physical inconsistency. The effect of the training data size on the RMSE of different models is shown in Table I. Here, the models are trained with 4, 6, 8, 10, and 20 experimental data points (i.e., 20%, 30%, 40%, 50%, and 100% of the available training data). The RMSE values of the coupled-physics model based on the sintering model developed by Gurrala et al. [8] and the new proposed one are 0.173 and 0.112, respectively, which are larger than the ML models because of the approximations used to represent the FFF process and bias in the model. The mean RMSE value of f^{pre}, which is trained with physics model input-output data consisting of 1525 input parameter combinations, is 0.362. The RMSE value of f^{pre} is greater than the RMSE of f^{phy} because the pre-trained model is an approximation of the physics model, which causes uncertainty and bias. The RMSE of the basic DNN model is about 0.025 when we use all the parts in the training set. The RMSE of the basic DNN model is better than f^{pre} and f^{phy} because experimental data are directly used as the training data for the basic DNN model, whereas f^{pre} and f^{phy} use the approximate physics model to generate either pre-training data or additional input to the ML model.

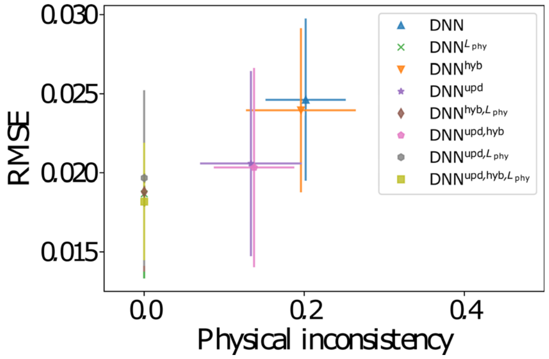
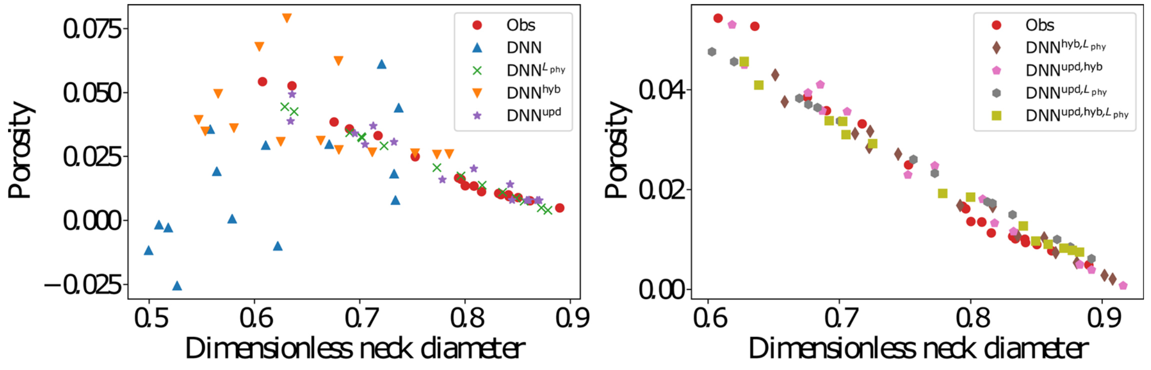


Fig. 6. Performance and physical inconsistency of proposed models.

The physics-based loss functions allow $\mathbf{DNN}^{\mathcal{L}_{\mathbf{phy}}}$ and $\mathbf{DNN}^{\mathbf{hyb},\mathcal{L}_{\mathbf{phy}}}$ to achieve physically meaningful results with a lower value of average RMSE than $\mathbf{DNN}^{\mathbf{hyb}}$ and improve the generalization performance. The $\mathbf{DNN}^{\mathbf{hyb}}$ model performs similarly to **DNN**, which shows that the physics model outputs do not improve the learning process significantly. The $\mathbf{DNN}^{\mathbf{upd}}$ and $\mathbf{DNN}^{\mathbf{upd},\mathbf{hyb}}$ models achieve a similar performance improvement as the DNN models that include physics-based loss functions. However, the models without physics constraints produce physically inconsistent results, as shown in Table I. The results show that pre-training the PIML model improves the performance, and the improvement is relatively larger as the amount of observed data gets smaller. Additionally, the models that are pre-trained ($\mathbf{DNN}^{\mathbf{upd}}$ and $\mathbf{DNN}^{\mathbf{upd},\mathbf{hyb}}$)



(a) **(b)**

Fig. 7. Comparison of model predictions with test data: (a) for the first four models and (b) for the last four models.

reach a physically more consistent initialization when they are updated with experimental data even without using physics constraints compared to models that are not pre-trained, **DNN** and $\mathbf{DNN}^{\mathbf{hyb}}$. The combination of all strategies ($\mathbf{DNN}^{\mathbf{upd},\mathbf{hyb},\mathcal{L}_{\mathbf{phy}}}$) allows the model to get closest to the ground truth.

The prediction accuracy (w.r.t. neck diameter and porosity) of different models trained with 100% of the experimental data is shown in Fig. 6. The x- and y-axes represent the physical inconsistency and the mean and standard deviation of RMSE, respectively. Figure 6 shows that models with physics-based loss functions produce physically consistent results. The incorporation of the physics knowledge using either the first or third strategy enables the models 2, 4–8 to generalize to configurations unseen in the training set.

To further analyze the improvement in model predictions, the predicted dimensionless overall neck diameter and porosity for the test set that comprises 19 FFF parts are visualized in Fig. 7. Porosity predictions of model 1 (**DNN**) have large physical inconsistency (Fig. 7a). For instance, the blue triangle with the lowest porosity prediction has a negative value (i.e., $(\hat{Y}_1, \hat{Y}_2) = (0.52, -0.025)$). Model 2 ($\mathbf{DNN}^{\mathcal{L}_{\mathbf{phy}}}$) predictions are physically consistent because of enforced physics constraints, but model 3 ($\mathbf{DNN}^{\mathbf{hyb}}$) predictions also do not follow a monotonic decreasing relationship. Models 4 and 6 ($\mathbf{DNN}^{\mathbf{upd}}$ and $\mathbf{DNN}^{\mathbf{upd},\mathbf{hyb}}$) have some physically inconsistent predictions. Figure 7b shows that $\mathbf{DNN}^{\mathbf{hyb},\mathcal{L}_{\mathbf{phy}}}$, $\mathbf{DNN}^{\mathbf{upd},\mathcal{L}_{\mathbf{phy}}}$, and $\mathbf{DNN}^{\mathbf{upd},\mathbf{hyb},\mathcal{L}_{\mathbf{phy}}}$ produce physically consistent model predictions, i.e., porosity and neck diameter predictions follow a monotonically decreasing relationship.

CONCLUSION

In this work, three strategies for physics-informed machine learning (PIML) are investigated for predicting the quality-related metrics of FFF-produced parts. First, a physics-based sintering model is developed to predict the overall average neck diameter and porosity of FFF parts, using the temperature evolution of filaments, material properties, part geometry, and process parameters as inputs. The developed sintering model offers two improvements over existing models: (1) consideration of realistic filament geometry and (2) allowing the filament geometry to change during the printing process. Next, several PIML models are developed to predict the bond quality and porosity of FFF parts by leveraging three strategies for incorporating physics knowledge into the DNN model: (1) physics-based loss functions, (2) using the outputs of the coupled multi-physics model as additional inputs to the DNN, and (3) pre-training a DNN with data generated using physics-based model and then updating it with experimental data. The physics-based loss functions exploit the relationship between bond quality and tensile strength of the FFF parts.

The numerical results show that the incorporation of physics knowledge not only improves the prediction accuracy while producing physically meaningful results, but also allows accurate model predictions even with smaller amounts of experimental data. Thus, the proposed approach helps to fill the physics knowledge gap in the ML model while leveraging the capability of ML to extract complex process-material-geometry relationships in AM and correcting for the approximation in the physics-based model.

Future work can include experimental data that consist of higher dimensional input, with multiple combinations, to further evaluate the performance of the proposed PIML strategies. Future work can also explore the generalization capabilities of the proposed strategies to parts of different geometry, as well as transfer learning to parts manufactured with different 3D printers and materials. Also, in the context of the third strategy, the data produced by experiments and physics models have different levels of credibility; thus the weighting of the two sources of data needs to be investigated in the future.

CONFLICT OF INTEREST

On behalf of all authors, the corresponding author states that there is no conflict of interest.

ELECTRONIC SUPPLEMENTARY MATERIAL

The online version of this article (https://doi.org/10.1007/s11837-020-04438-4) contains supplementary material, which is available to authorized users.

REFERENCES

1. C. Bellehumeur, L. Li, Q. Sun, and P. Gu, *J. Manufact. Process.* 6(2), 170–178 (2004). https://doi.org/10.1016/S1526-6125(04)70071-7.
2. F. Chollet et al. Keras (2015). https://github.com/fchollet/keras.
3. S. Costa, F. Duarte, and J. Covas, *J. Mater. Process. Technol.* 245, 167–179 (2017). https://doi.org/10.1016/j.jmatprotec.2017.02.026.
4. T. DebRoy, W. Zhang, J. Turner, and S. Babu, *Scripta Materialia* 135, 119–124 (2017). https://doi.org/10.1016/j.scriptamat.2016.12.005.
5. D. Ding, Z. Pan, D. Cuiuri, H. Li, S. van Duin, and N. Larkin, *Robot. Comput. Integrated Manufact.* 39, 32–42 (2016). https://doi.org/10.1016/j.rcim.2015.12.004.
6. J. Frenkel, *J. Phys.* 9, 385 (1945).
7. S. Garzon-Hernandez, D. Garcia-Gonzalez, A. Jérusalem, and A. Arias, . *Mater. Des.* 188, 108,414 (2020). https://doi.org/10.1016/j.matdes.2019.108414.
8. P.K. Gurrala, and S.P. Regalla, *Virtual Phys. Prototyp.* 9(3), 141–149 (2014). https://doi.org/10.1080/17452759.2014.913400.
9. R.W. Hopper, *J. Am. Ceram. Soc. (United States)* (1984). https://doi.org/10.1111/j.1151-2916.1984.tb19692.x.
10. X. Jia, J. Willard, A. Karpatne, J.S. Read, J.A. Zwart, M. Steinbach, and V. Kumar, Physics-guided machine learning for scientific discovery: an application in simulating lake temperature profiles (2020). arXiv:2001.11086.
11. B. Kapusuzoglu, M. Sato, S. Mahadevan, and P. Witherell, *J. Manufact. Sci. Eng.* (2020). https://doi.org/10.1115/1.4048073.
12. A. Karpatne, W. Watkins, J. Read, and V. Kumar, Physics-guided neural networks (PGNN): an application in lake temperature modeling (2017). arXiv:1710.11431.
13. M. Khanzadeh, S. Chowdhury, M. Marufuzzaman, M.A. Tschopp, L. Bian, *J. Manufact. Syst.* 47, 69–82 (2018). https://doi.org/10.1016/j.jmsy.2018.04.001.
14. O. Kwon, H.G. Kim, W. Kim, G.H. Kim, and K. Kim, A convolutional neural network for prediction of laser power using melt-pool images in laser powder bed fusion. IEEE Access (2020).
15. Y. Ling, J. Mullins, and S. Mahadevan, *J. Comput. Phys.* 276, 665–680 (2014). https://doi.org/10.1016/j.jcp.2014.08.005.
16. P. Nath, Z. Hu, and S. Mahadevan, *Modell. Simul. Mater. Sci. Eng.* 27(4), 044,003 (2019). https://doi.org/10.1088/1361-651x/ab1676.
17. O. Pokluda, C.T. Bellehumeur, and J. Vlachopoulos, *AIChE J.* 43(12), 3253–3256 (1997). https://doi.org/10.1002/aic.690431213.
18. N. Rosenzweig, and M. Narkis, *Polym. Eng. Sci.* 21(17), 1167–1170 (1981). https://doi.org/10.1002/pen.760211709.
19. C.A. Schneider, W.S. Rasband, and K.W. Eliceiri, *Nature Methods* 9(7), 671–675 (2012).
20. B. Zhang, S. Liu, and Y.C. Shin, *Addit. Manufact.* 28, 497–505 (2019). https://doi.org/10.1016/j.addma.2019.05.030.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.