\title{Global sensitivity analysis of physics-informed machine learning methods}

%% Group authors per affiliation:

\author{Berkcan Kapusuzoglu, Sankaran Mahadevan\corref{corauthor}}

\address{Department of Civil and Environmental Engineering, Vanderbilt University, TN 37235}

\cortext[corauthor]{Corresponding author\\Phone: 1-615-322-3040. Fax: 1-615-322-3365. Postal address: Vanderbilt University, VU Station B 356077, Nashville, TN 37235-6077, United States.\\ \emph{E-mail address}: sankaran.mahadevan@vanderbilt.edu (S. Mahadevan).}

\date{\today}

\begin{frontmatter}

\begin{abstract}

Predicting the effect of inputs to the uncertainty in the output in the FFF process is essential for achieving proactive quality control of manufactured parts. Variance-based sensitivity analysis based on Sobol' indices is used to quantify the relative contributions of different uncertainty sources to the uncertainty in the porosity of FFF parts.

Incorporating physics knowledge into a deep neural network (DNN) can reduce the computational effort. The inclusion of physics knowledge into the DNN model not only improved the model accuracy, but also allowed to estimate Sobol indices with smaller amount of experimental data.

Uncertainty in model predictions is considered by implementing Monte Carlo dropout in the DNN model. This allows us to derive prediction intervals for the Sobol index estimates.

\end{abstract}

% \vspace{.3cm}

\begin{keyword}

Global sensitivity analysis\sep Sobol' index\sep Additive manufacturing\sep Fused filament fabrication\sep Physics-informed machine learning\sep Deep learning

\end{keyword}

\end{frontmatter}

\linenumbers

% Introduction

\section{Introduction}

Fused filament fabrication (FFF), an extrusion-based deposition technique, is a widely used additive manufacturing (AM) process. Among other rapid prototyping technologies, FFF is popular due to its low cost, easy operation, and suitability for complex geometries. The materials are joined layer upon layer by extruding a molten filament through a heated nozzle, which moves in the horizontal plane onto a build plate or onto other filaments that move in the vertical direction. As the material is deposited, it cools down, solidifies and bonds with the surrounding filaments. The bond quality and porosity of a part is dependent on the temperature history at the interfaces between filaments. Thus, it is important to predict the temperature evolution of filaments for estimating the final mesostructure of the printed part.

In recent years, research efforts have focused on model-based methods for the selection of input variables of the AM. Although, physics-based models do not require large amounts of data, they are generally limited by their computational complexity or incomplete physics. Further, the estimation of Sobol indices require a significant number of model runs; thus the use of physics-based modeling in AM has been challenging and limited. Nowadays, several studies have built experimental data-driven machine learning (ML) models to predict the quantity of interest (QoI). Generally, the construction of ML models does not require in-depth knowledge of the complex physics in the AM process. ML models can learn complex systems using available observations, but the accuracy of these models depends on the quality and quantity of the experimental data. If the available data is limited, then the true nature of the process cannot be captured. Further, since ML model predictions do not consider physical laws, they can produce physically inconsistent results.

Karpatne et al.~\cite{karpatne2017physics} 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. In this work, we investigated a similar approach by incorporating the physics knowledge into the ML models to better capture the physics of the AM process by leveraging physical laws while improving the generalization performance of data-driven AM models. Eight different physics-informed machine learning (PIML) models are developed to predict the bond quality and porosity of FFF parts through combinations of three strategies: (1) incorporating physics constraints within the loss function of the DNN, (2) using the coupled multi-physics model outputs as additional inputs to the DNN model, and (3) pre-training a DNN model with physics model input-output and then updating it with experimental data. The proposed methods use multiple physics-based loss functions and address two different physical QoIs (neck diameter and porosity) for enhancing the experimental data-driven ML models for AM. The physics constraints exploit the relationship between porosity and tensile strength and physical laws related to the mesostructure of FFF parts.

In order to identify the relative contribution of model inputs to the uncertainty in the model output (porosity), the AM process needs to be repeated by running multiple experiments with different values of process parameters. This is practically impossible; thus physics-based modeling approaches are used to estimate Sobol indices. However, the computation of Sobol indices require a significant number of Monte Carlo simulations; thus it is not affordable when the physics models are computationally expensive.

To handle this problem, we built eight different PIML models to approximate the physics of the AM process and to estimate the Sobol indices with less computational effort and experimental data than a Gaussian process model.

The various sources of uncertainty are classified as (a) Epistemic Uncertainty due to lack of knowledge, and (b) Aleatory Uncertainty due to the inherent variability in the system. Global sensitivity analysis (GSA)~\cite{saltelli2008global} provides a quantitative assessment of the relative contribution of model inputs to the uncertainty in the model output. GSA techniques can be either data-driven (i.e., based on analysis of variance ANOVA) or model-based (e.g., the computation of Sobol indices~\cite{sobol2001global}). One of the main focus of this paper is to assess the relative contributions of each uncertainty sources. Model outputs can have uncertainty even for a fixed input model-based prediction when there exists uncertainty about the model. Moreover, inputs of the model can be deterministic or random and epistemic uncertainty can be present in both cases because of lack of data. When the input is deterministic and its value is unknown, then there is only epistemic uncertainty. Whereas, if the input is a random variable with an unknown distribution type and/or parameter, then there would be both aleatory and epistemic uncertainty. The epistemic uncertainties in model inputs contribute additional uncertainty in the output of the model.

Gal and Ghahramani~\cite{gal2016dropout} showed that performing approximate variational inference is equivalent to Monte Carlo (MC) dropout, which infers the posterior by performing dropout not only while training a model but also at test time. The randomly chosen neurons are temporarily removed from the network along with its connections. Next, the gradients of neurons weights are calculated on a thinner neural network for each training data and these gradients are then averaged over the training sets in the mini-batch to obtain the weights of overall network. The optimization of Bayesian neural networks with the MC dropout is equivalent to using dropout as regularization on neural networks. However, in contrast to standard neural networks, the MC dropout performs dropout and generates random samples following a Bernoulli distribution for each neuron in the input and hidden layers during testing. The dropout is applied to the neuron that takes the value of $0$ with a given dropout probability $p\_d$. The outputs of the network are predicted using the collection of generated random samples from the posterior predictive distribution and the uncertainty in the prediction of a new data is quantified with the trained network. Although there have been considerable discussions on drawbacks of the MC dropout strategy and approximated posteriors with the dropout variational inference often show inaccurate results~\citep{kuleshov2018accurate,gal2018sufficient,osband2018randomized}, MC dropout networks have several promising features. For example, the computational effort of the optimization of Bayesian neural networks with the MC dropout is comparable to the standard neural networks. Moreover, the simplicity of the MC dropout strategy provides an efficient way of Bayesian inference to quantify the model prediction uncertainty with variety of neural networks, such as feedforward neural networks, convolutional neural networks, and recurrent neural networks.

In summary, the contributions of this paper are as follows:

….

The outline of the rest of this paper is as follows. Section~\ref{Sec:Methods} presents the proposed methodology .... The proposed methodology is illustrated for a numerical example in Section~\ref{Sec:Results}. Concluding remarks are provided in Section~\ref{sec:conclusion}.

% Methodology

\section{Materials and Methods}\label{Sec:Methods}

The proposed methodology for process parameter optimization under uncertainty consists of the following steps:

\begin{enumerate}

\item Probabilistic sensitivity analysis

\item Surrogate modeling of physics models

\end{enumerate}

The following subsections describe these steps in detail.

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

\subsection{Probabilistic sensitivity analysis}\label{Sec:Sensitivity}

A probabilistic sensitivity analysis, commonly referred to as global sensitivity analysis (GSA) is used to assess the relative contribution of each uncertainty source towards the uncertainty of the model output (bond length in this case). Model inputs or parameters with negligible contribution can be fixed at their mean values in order to reduce the number of stochastic variables. Variance-based GSA, using Sobol' indices, is adopted in this paper, as briefly described below.

\subsubsection{Variance-based global sensitivity analysis}

Consider a real integrable deterministic one-to-one system response function $\mathrm{Y}=f(\BoldMath{X})$, where $\BoldMath{X}=\{\mathrm{X}\_1, ..., \mathrm{X}\_k\}$ are mutually independent model inputs, $f(\boldsymbol{\cdot})$ is the computational model and $\mathrm{Y}$ is the model output. As shown by \cite{sobol2001global}, the variance of $\mathrm{Y}$ can be decomposed as

\begin{align}

V(\mathrm{y}) = & \sum\_i^k V\_i + \sum\_{i\_1}^k\sum\_{i\_2=i\_1+1}^k V\_{i\_1i\_2} + \nonumber \\

& \sum\_{i\_1}^k\sum\_{i\_2=i\_1+1}^k\sum\_{i\_3=i\_2+1}^k V\_{i\_1i\_2i\_3} + ... + V\_{12...k}

\label{eq:SobolVar}

\end{align}

where $V\_i$ is the variance of $\mathrm{Y}$ caused by only $\mathrm{X\_i}$, $V\_{i\_1...i\_p} (p\geq2)$ indicates the variance of $\mathrm{Y}$ caused by the interactions of $\{\mathrm{X\_{i\_1}},...,\mathrm{X\_{i\_p}}\}$.

The Sobol indices are defined by dividing both sides of Eq.~\eqref{eq:SobolVar} with $V(\mathrm{Y})$

\begin{align}

1= & \sum\_i^k S\_i + \sum\_{i\_1}^k\sum\_{i\_2=i\_1+1}^k S\_{i\_1i\_2} + \nonumber \\

& \sum\_{i\_1}^k\sum\_{i\_2=i\_1+1}^k\sum\_{i\_3=i\_2+1}^k S\_{i\_1i\_2i\_3} + ... + S\_{12...k}

\label{eq:SobolIndex}

\end{align}

where $S\_i$ is the first-order or main effects index that assess the contribution of $\mathrm{X\_i}$ individually to the variance of the output $\mathrm{Y}$ without considering the interactions with other inputs. The variables with higher first-order index are the important ones to be accounted for. Other indices $S\_{i\_1...i\_p} (p\geq2)$ in Eq.~\eqref{eq:SobolIndex} are higher-order indices that measure the contribution of the interactions of $\{\mathrm{X\_{i\_1}},...,\mathrm{X\_{i\_p}}\}$.

In other words, the evaluation of $S\_i$ is as follows:

\begin{equation}

S\_i = \frac{V\_i}{V(\mathrm{Y})} = \frac{V\_{X\_i}(E\_{\BoldMath{X}\_{-i}}(\mathrm{Y}|\mathrm{X}\_i))}{V(\mathrm{Y})}

\label{eq:FirstIndex}

\end{equation}

where $\BoldMath{X}\_{-i}$ are all the model inputs other than $\mathrm{X}\_i$.

The overall contribution of $\mathrm{X\_i}$ considering both individual effect and interactions with other inputs is measured by the total effects index $S\_i^T$:

\begin{equation}

S\_i^T = 1 - \frac{V\_{-i}}{V(\mathrm{Y})} = \frac{V\_{\BoldMath{X}\_{-i}}(E\_{\mathrm{x}\_i}(\mathrm{Y}|\BoldMath{X}\_{-i}))}{V(\mathrm{Y})}.

\label{eq:TotalIndex}

\end{equation}

The calculation of first-order and total effects indices requires a deterministic function. Furthermore, the total effects index $S\_i^T$ is only meaningful for uncorrelated model inputs \citep{saltelli2002relative}. Whereas, the first-order index $S\_i$ can be calculated for both correlated and uncorrelated model inputs~\citep{saltelli2002relative}.

The computation of $S\_i$ analytically is nontrivial since $E\_{\BoldMath{X}\_{-i}}(\boldsymbol{\cdot})$ requires multi-dimensional integrals. Using Monte Carlo simulation (MCS) to measure $S\_i$ is also expensive because calculation of the numerator of $S\_i$ requires a double-loop MCS, where the inner loop $E\_{\BoldMath{X}\_{-i}}(\mathrm{Y}|\mathrm{X}\_i)$ computes the mean value of $\mathrm{Y}$ using $n\_1$ random samples of $\BoldMath{X}\_{-i}$ and the outer loop computes $V\_{X\_i}(E\_{\BoldMath{X}\_{-i}}(\mathrm{Y}|\mathrm{X}\_i))$ by iterating the inner loop $n\_2$ times for different values of $\mathrm{X}\_i$. Moreover, the computation of $V(\mathrm{Y})$ requires additional $n\_3$ MCS iterations. The total computational cost of the first-order and total effect indices (i.e. the number of function evaluations, $N\_f$) is approximately $N\_f = kN^2+N$, where $N=n\_1=n\_2=n\_3$. Thus, the calculation of indices become unaffordable for a large number of inputs and expensive models since the required number of random samples are of the order greater than 1000 in many practical applications.

The sampling-based method modularized global sensitivity analysis (MGSA) proposed by \cite{li2016efficient}, which has a computational cost that is not proportional to the model input dimension, can directly estimate the first-order Sobol' index from Monte Carlo samples with a single-loop instead of the double-loop MCS. The first-order Sobol' index can be computed by dividing the input variable $X\_i$ into equally probable intervals $\boldsymbol{\phi}={\phi^1,...,\phi^M}$:

\begin{equation}

S\_i = 1 - \frac{E\_{\phi}(V\_{\phi^p}(Y))}{V(Y)},\ p=1,...,M

\end{equation}

where $V\_{\phi^p}(Y)$ is the variance of $Y$ when $X\_i$ in the subspace $\phi^p$, $V(Y)$ represents the variance of the system response $Y$. The main advantages of this efficient data-driven method based on the concept of stratified sampling~\citep{iooss2009global} are that the computational cost is not proportional to the number of model inputs and the physics/computational model does not have to be available since the input-output samples are enough to compute the first-order Sobol index. Further advances in this direction based on importance sampling and kernel functions have been reported by \cite{decarlo2018efficient}.

An important challenge is the computation of Sobol indices. The computation of $S\_i$ analytically is nontrivial since $E\_{\BoldMath{X}\_{-i}}(\boldsymbol{\cdot})$ requires multi-dimensional integrals. Using Monte Carlo simulation (MCS) to measure $S\_i$ is also expensive because calculation of the numerator of $S\_i$ requires a double-loop MCS, where the inner loop $E\_{\BoldMath{X}\_{-i}}(\mathrm{Y}|\mathrm{X}\_i)$ computes the mean value of $\mathrm{Y}$ using $n\_1$ random samples of $\BoldMath{X}\_{-i}$ and the outer loop computes $V\_{X\_i}(E\_{\BoldMath{X}\_{-i}}(\mathrm{Y}|\mathrm{X}\_i))$ by iterating the inner loop $n\_2$ times for different values of $\mathrm{X}\_i$. Moreover, the computation of $V(\mathrm{Y})$ requires additional $n\_3$ MCS iterations. The total computational cost of the first-order and total effect indices (i.e. the number of function evaluations, $N\_f$) is approximately $N\_f = kN^2+N$, where $N=n\_1=n\_2=n\_3$. Thus, the calculation of indices become unaffordable for large number of inputs and expensive models since the required number of random samples are of the order greater than 1000 in many practical applications.

The methodologies that have been used to reduce the computational cost of the Sobol indices can be categorized into analytical and sample-based methods. In the former one, the original computational model $f(\boldsymbol{\cdot})$ is generally approximated by a surrogate model in order to convert the multi-dimensional integral into multiple uni-variate integrals. Whereas, the latter one, which has been explained earlier, generates samples such as the double-loop MCS to evaluate the indices. The computational cost of most sample-based methods is proportional to the number of model input $k$. However, the method developed by Li~\cite{li2016efficient} is one of those sample-based methods which has a computational cost that is not proportional to the model input dimension such as the improved FAST method~\cite{tarantola2006random}.

The sample-based method modularized global sensitivity analysis (MGSA) proposed by Li~\cite{li2016efficient} can directly estimate the Sobol index from Monte Carlo samples without evaluating the physics/computational model multiple times. The conditional variance and mean in the first-order Sobol index can be computed at an unknown but existing location of model inputs. The proposed method separates index calculations for different model inputs and model input sampling, model and index calculations are separate processes. The main advantages of this proposed method are that the computational cost is not proportional to the number of model inputs and the physics/computational model does not have to be available since the input-output samples are enough compute the first-order Sobol index. The method is can be represented by two algorithms and the only difference between these two algorithms is that Algorithm 1 first computes the inner loop $E\_{\boldsymbol{X}\_{-i}}(YX\_i)$, then the outer loop $V\_{\boldsymbol{X}\_i}(\cdot)$, whereas Algorithm 2 first computes the inner loop $V\_{\boldsymbol{X}\_{-i}}(YX\_i)$, then the outer loop $E\_{\boldsymbol{X}\_i}(\cdot)$. The number of samples needed to guarantee accurate results is 2500 samples according to~\cite{li2016efficient}. In this paper, 10000 samples and 100 number of intervals (divided equal length CDF of model inputs) are generated to reach desirable accuracy of the computed index.

\subsubsection{Uncertainty Quantification in Bayesian Neural Network}

The various sources of uncertainty are classified as (a) Epistemic Uncertainty due to lack of knowledge, and (b) Aleatory Uncertainty due to the inherent variability in the system. Epistemic uncertainty is caused by the unknown values of the model parameters such as neuron weights. The BNN is used to describe the epistemic uncertainty caused by the model by placing distributions over the model parameters (neuron weights). Whereas the aleatory uncertainty is caused by the natural variability in the QoI. For example, the AM parts printed at the same process parameters may fail at different tensile loads. Thus, the aleatory uncertainty is not a property of the model but of the data. The observation noise parameter $\sigma$ needs to be tuned in order to capture aleatory uncertainty. A model developed by \cite{kendall2017uncertainties} allows studying the effects of modeling aleatoric and epistemic uncertainty separately or modeling both uncertainties together.

In standard neural networks, the observation noise $\sigma$ is often fixed as part of the model's weight decay. On the other hand, it can be learned as a function of the data by fixing a Gaussian likelihood as the minimization objective:

\begin{equation}

\mathcal{L}\_{BNN}(\theta) = \frac{1}{N}\sum\_i \frac{1}{2} \hat{\sigma}^{-2}\_i ||\mathbf{y}\_i - \hat{\mathbf{y}}\_i ||^2 + \frac{1}{2}\text{log}\ \hat{\sigma}^{-2}\_i + \lambda g(\mathbf{w})

\end{equation}

where $N$ is the number of observations $\mathbf{y}\_i$, $\hat{\mathbf{y}}\_i$ and $\hat{\sigma}^{2}\_i$ are the predictive mean and variance corresponding to input indexed by $i$, $\lambda$ is the weight decay parameter and $g(\mathbf{w})$ represents a regularization function (or the penalty function which is often $L\_2$ regularization). The neuron weights can be drawn from the approximate posterior $\hat{\mathbf{w}}\sim q(\mathbf{w})$ to obtain the model outputs of a BNN denoted as $\mathbf{f}^{\hat{w}}(\mathbf{x}) = [\hat{\mathbf{y}}, \hat{\sigma}^{2}]$. The second part of the loss function can be regarded as an uncertainty regularization term and it prevents the network from predicting infinite uncertainty. The network learns the observation noise implicitly from the loss function. In order to have a numerically stable network during training, the log variance $\text{log}\ \hat{\sigma}^{2}\_i$ is predicted instead of predicting $\hat{\sigma}^{2}\_i$.

The predictive mean and predictive uncertainty are estimated by collecting the results of stochastic forward passes through the model. The mean prediction of the model with $T$ MC samples can be approximated by

\begin{equation}\label{eq:BNNpredmean}

\mathbb{E}(\mathbf{y}) \approx \frac{1}{T} \sum\_{t=1}^{T} \mathbf{f}^{\hat{w}}(\mathbf{x}).

\end{equation}

\subsection{PIML for additive manufacturing}\label{sec:method3}

Although physics-based models predicting the temperature evolution, bond formation and mesostructure evolution of FFF parts are based on physical laws, they introduce bias due to 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. 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~\cite{willard2020integrating}. 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 trained with experimental data, (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.

\subsubsection{Physics-informed loss functions}

A direct strategy to improve ML model predictions is by including physics-based loss functions~\cite{karpatne2017physics}. Consider a PIML model with inputs $\boldsymbol{X}$ and outputs $\boldsymbol{\hat{Y}}$ trained using physical laws that are incorporated as constraints into the loss function:

\begin{align}\label{eq:lossfunc}

\mathcal{L} = \mathcal{L}\_{\rm DNN}(\boldsymbol{Y}, \boldsymbol{\hat{Y}}) + \lambda\_{\rm phy}\mathcal{L}\_{\rm phy}(\boldsymbol{\hat{Y}}),

\end{align}

where $\mathcal{L}\_{\rm DNN}$ is the regular training loss of a DNN that evaluates a supervised error (e.g., root mean squared error (RMSE); $\mathcal{L}\_{\rm DNN}(\boldsymbol{Y}, \boldsymbol{\hat{Y}})=\sqrt{\sum\_{i=1}^{n} (Y\_{i}- \hat{Y}\_{i})^2/n}$), which measures how far off the predictions $\boldsymbol{\hat{Y}}$ are from the observations $\boldsymbol{Y}$ for the $n$ training samples, and $\mathcal{L}\_{\rm phy}$ is the physics-based loss function, whose contribution is controlled by a hyperparameter $\lambda\_{\rm phy}$. The inclusion of $\mathcal{L}\_{\rm phy}$ ensures physically consistent model predictions and can decrease the generalization error even when there is a small amount of training data~\cite{karpatne2017physics}. In addition, $\mathcal{L}\_{\rm phy}$ does not require experimental observations; the data obtained from the physics model is used to evaluate physics-based loss functions.

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

\begin{align}

& {}\mathcal{L}\_{{\rm phy}, 1}(\boldsymbol{\hat{Y}}) = \frac{1}{N} \sum\_{i=1}^N \rm{ReLU}(-\mathnormal{\hat{Y}}\_{1,i}), \notag\\

& {}\mathcal{L}\_{{\rm{phy}}, 2}(\boldsymbol{\hat{Y}}) = \frac{1}{N} \sum\_{i=1}^N \rm{ReLU}(\mathnormal{\hat{Y}}\_{1,i}-\mathnormal{d\_{max}}), \notag\\

& {}\mathcal{L}\_{{\rm phy}, 3}(\boldsymbol{\hat{Y}}) = \frac{1}{N} \sum\_{i=1}^N \rm{ReLU}(-\mathnormal{\hat{Y}}\_{2,i}), \notag\\

& {}\mathcal{L}\_{{\rm phy}, 4}(\boldsymbol{\hat{Y}}) = \frac{1}{N} \sum\_{i=1}^N \rm{ReLU}(\mathnormal{\hat{Y}}\_{2,i}-\phi\_{0}), \notag\\

& {}\mathcal{L}\_{{\rm phy}, 5}(\boldsymbol{\hat{Y}}) = \frac{1}{N} \sum\_{i=1}^N \rm{ReLU}(\Delta\_i),

\end{align}

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 which are greater than the maximum dimensionless neck diameter $d\_{max}=1$. The fourth loss function penalizes the model when porosity predictions $\mathnormal{\hat{Y}}\_{2,i}$ are greater than initial porosity $\phi\_0$ of $i$th 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 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, $\mathnormal{\hat{Y}}\_{1,\rm{sorted}}$, and dimensionless neck diameter predictions corresponding to sorted tensile strength estimates ($\sigma\_{TS}(\mathnormal{\hat{Y}}\_{2,\rm{sorted}}, \boldsymbol{\xi})$), $\mathnormal{\hat{Y}^{'}}\_{1,i}$, i.e., $\Delta\_i = \mathnormal{\hat{Y}}\_{1,\rm{sorted},i} - \mathnormal{\hat{Y}^{'}}\_{1,i}$. The maximum stress for longitudinal raster orientation, $\sigma\_{TS}(\mathnormal{\hat{Y}}\_{2}, \boldsymbol{\xi})$ with $\boldsymbol{\xi}=\{\sigma\_{01}, \sigma\_{02}, C\_{\sigma}\}$ being the material parameters, is computed according to the analytical expression proposed in Garzon et al.~\cite{garzon2020design} using the porosity predictions:

\begin{align}

\sigma\_{TS} = \sigma\_{01} \bigg[\rm{exp}\Big( (1-\mathnormal{\hat{Y}}\_{2})^{C\_{\sigma} n\_l } \Big) - \mathnormal{\hat{Y}}\_{2} \bigg] + \sigma\_{02}(1-\mathnormal{\hat{Y}}\_{2}),

\end{align}

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 (i.e., overall bond quality) 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 $i$th 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 $(i+1)$th part has less voids than $i$th 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.

\subsubsection{Pre-trained PIML model}\label{sec:method4}

In AM, especially in the FFF process with not a 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 is important for good quality model predictions in AM.

In order 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 due to lack of knowledge of 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~\cite{jia2020physics}.

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 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 Section~\ref{sec:results}, the pre-training strategy exercises the physics model over 1525 input combinations, whereas the second strategy above only employs the physics model over 39 input combinations, which are the experimental inputs. 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). Further, 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 much faster (i.e., with less epochs) and with less samples.

The three proposed strategies to predict the QoIs (neck diameter and porosity) are shown in Fig.~\ref{fig:diagram}. Figure~\ref{fig:diagram}(a) shows the first method, where the physical knowledge is included through constraints within the loss function of a DNN trained with experimental data. Figure~\ref{fig:diagram}(b) shows the second method, where the outputs of the physics model are additional inputs to the DNN model. Figure~\ref{fig:diagram}(c) shows the third method, where a DNN model is first 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.

%%======================================================

%% Figure

%%======================================================

\begin{figure}[!ht]

\centering

\includegraphics[width=0.8\textwidth]{Fig4.pdf}

\caption{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.}

\label{fig:diagram}

\end{figure}

%======================================================

\subsubsection{Combination of PIML strategies}\label{sec:method5}

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

\begin{multicols}{2}

\begin{enumerate}

\item $\mathbf{\rm DNN}$

\item $\mathbf{\rm DNN^{\mathcal{L}\_{\rm phy}}}$

\item $\mathbf{\rm DNN^{\rm upd}}$

\item $\mathbf{\rm DNN^{\rm upd, \mathcal{L}\_{\rm phy}}}$

\end{enumerate}

\end{multicols}

In model 1, a deep neural network $\mathbf{\rm DNN}$ 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; and the outputs are overall dimensionless neck diameter and porosity. These inputs and outputs are the same as those used in the physics-based model $f^{\rm phy}$. Model 2 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.~\eqref{eq:lossfunc}, $\mathbf{\rm DNN^{\mathcal{L}\_{\rm phy}}}$. Model 3 pursues the second strategy: a hybrid physics-based neural network $\mathbf{\rm DNN^{\rm hyb}}$ is trained by using the outputs $\boldsymbol{\hat{Y}}^{\rm phy}$ of the physics-based model $f^{\rm phy}$ as extra inputs in addition to $\boldsymbol{X}$, i.e., $\boldsymbol{X}^{\rm hyb}=[\boldsymbol{X}, \boldsymbol{\hat{Y}}^{\rm phy}]$. Model 4 ($\mathbf{\rm DNN^{\rm upd}}$) pursues the third strategy, where a pre-trained DNN model $f^{\rm pre}$ (which is trained with the coupled multi-physics model input-output described in Section~\ref{sec:method4}) is updated with experimental data. 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. Thus, model 8 ($\mathbf{\rm DNN^{\rm upd, hyb, \mathcal{L}\_{\rm phy}}}$) combines the use of physics model outputs as additional inputs to the updated DNN model $\mathbf{\rm DNN^{\rm upd}}$ (which results in $\mathbf{\rm DNN^{\rm upd, hyb}}$) and the physics constraints $\mathcal{L}\_{\rm phy}$ are also incorporated within the loss function of the DNN.

\subsection{Experimental work}

A commercial material, Ultimaker Black ABS, was used in the experiments. A unidirectional and aligned building strategy was adopted at a specified printer nozzle temperature and extrusion speed. Two different options for the deposition sequence of the filaments were considered, as shown in Fig.~\ref{fig:DeposSeqSequential}. In Fig.~\ref{fig:DeposSeqSequential}-(a), the filaments are sequenced from left to right in all the layers. In Fig.~\ref{fig:DeposSeqSequential}-(b), the filaments are sequenced from left to right in odd numbered layers and from right to left in even numbered layers. The filament numbers in the two figures correspond to the two deposition sequence options.

%%======================================================

%% Figure

%%======================================================

\begin{figure}[H]

\centering

\begin{subfigure}{0.49\textwidth}

\includegraphics[width=\textwidth]{FIGs/Filaments\_6\_15.pdf}

\caption{}

\end{subfigure}

%

\begin{subfigure}{0.49\textwidth}

\includegraphics[width=\textwidth]{FIGs/Filaments\_6\_15\_FollowingSequential.pdf}

\caption{}

\end{subfigure}

\captionsetup{width=0.90\textwidth}

\caption{Deposition sequence of unidirectional 90 filaments:~(a) from left to right for all the layers and~(b) from left to right in odd numbered layers and from right to left in even numbered layers}

\label{fig:DeposSeqSequential}

\end{figure}

%%======================================================

Multiple rectangular-shaped specimens were produced with the same geometry but different combinations of process parameter values. For each specimen, the temperature distribution at the top of each layer during deposition was monitored using an infrared thermography camera. Thermal images were recorded with a specified frequency until all filaments were deposited. The neck growth between the filaments and the total void area of the parts were identified at a specified cross-section with the use of microscopy images processed through the ImageJ software~\citep{schneider2012nih}. The statistical properties of the neck growth along the length of the specimens were constant. Therefore, all specimens were sectioned at the midpoint to analyze the mesostructural feature of interest only at that cross-section.

% Numerical results

\section{Numerical results and discussion}\label{Sec:Results}

The experimental setup used to build rectangular acrylonitrile butadiene styrene (ABS) amorphous polymer specimens of length 35 mm, width 12 mm, and thickness 4.2 mm is shown in Fig.~\ref{fig:ExpSetup}. The specimens were created on an Ultimaker 2 extended+ printer, which is within an enclosure to reduce the part variability; a commercial material, Ultimaker Black ABS, was used. All parts were printed through a nozzle with 0.8 mm diameter. The build plate temperature was constant and set to $110^{\circ}$C. The extrusion rate and vertical position of the nozzle were adjusted by the printer to be able to produce each filament with 0.8 mm width and 0.7 mm height.

%%======================================================

%% Figure

%%======================================================

\begin{figure}[H]

\centering

\includegraphics[width=0.5\textwidth]{FIGs/exp\_setup.pdf}

\caption{The experimental setup}

\label{fig:ExpSetup}

\end{figure}

%%======================================================

The surface temperature profiles of extruded filaments were monitored using an infrared thermography camera as shown in Fig.~\ref{fig:ExpSetup}. The extrusion of the next layer prevents the camera from monitoring the temperature profiles of the previous layers. Due to the inability to obtain temperature data of the filaments below the top layer, we could only predict the quality of the intra-layer bonding using temperature profiles of extruded filaments within that layer. Thermal images were recorded with a frequency of 10 Hz until the deposition of all filaments was completed.

All specimens used in this study are produced with unidirectional filaments to enhance the effects of process parameters on the bond quality between adjoining filaments. Each filament of the rectangular part is deposited at a specified printer nozzle temperature $T\_n$ and extrusion speed $v\_p$. The temperature evolution of the interfaces, and the neck growth between the filaments (the mesostructural feature of interest), are predicted at $z\_{\rm cut}=L/2$ as shown in Fig.~\ref{fig:DeposSeqSequential}. The process parameters and material properties used in this work are presented in Table~\ref{table:Parameters}. The specific heat capacity and density of the material are calibrated together as a single term $\alpha=\rho C$, where $\rho$ and $C$ are density (kg/m$^3$) and specific heat capacity (J/kg $^{\circ}$C) respectively. The analysis assumes temperature dependent material properties such as material viscosity $\eta$ and surface tension $\Gamma$. The surface tension of ABS P400 at 240$^{\circ}$C is 0.029 N/m as reported by \cite{bellehumeur2004modeling} with a temperature dependence $\Delta\Gamma/\Delta T = -\gamma$ N/m $\cdot$ K, where the neck growth model parameter $\gamma=0.00345$. The temperature dependent material viscosity $\eta$ is given by $\eta=\eta\_r\ \text{exp}[-\beta (T-T\_r)]$, where the material viscosity at the reference temperature ($T\_r=240^{\circ}$C) $\eta\_r$ is 5100 Pa $\cdot$ s,\ $\beta$ is a model parameter that is selected as 0.056 by \cite{sun2008effect}, and $T$ is the temperature of the material at a given time instance with units Kelvin (K).

\begin{table}[H]

\centering

\captionsetup{width=0.90\textwidth}

\caption{Process parameters and material properties}

\label{table:Parameters}

\resizebox{0.8\textwidth}{!}{

\begin{tabular}{l\*{1}{r}}

\hline

{Property}& Value \\

\hline

Printer nozzle temperature ($^{\circ}$C) & 240 \\

Build plate temperature ($^{\circ}$C) & 110 \\

Printer extrusion speed (m/s) & 0.042 \\

Filament length (m) & 0.035 \\

Filament width (m) & 0.008 \\

Filament thickness (m) & 0.007 \\

Fraction of filament's perimeter for all contacts & 0.15 \\

Convective heat transfer coefficient (W/m$^2$ $^{\circ}$C) & 86\\

Conductive heat transfer coefficient between filaments (W/m$^2$ $^{\circ}$C) & 200\\

Conductive heat transfer coefficient between filament and build plate (W/m$^2$ $^{\circ}$C) & 86\\

Thermal conductivity (W/m $^{\circ}$C) & 0.15\\

$\alpha$ (J/m$^3$ $^{\circ}$C) & 1.197110 $\times 10^{6}$\\

\hline

\end{tabular}

}

\end{table}

% Conclusion

\section{Discussion and conclusions}\label{Sec:Conclusion}