

DETC2021-70557

**DRAFT: STOCHASTICALLY-TRAINED PHYSICS-INFORMED NEURAL NETWORKS:
APPLICATION TO THERMAL ANALYSIS IN METAL LASER POWDER BED FUSION**

Justin Pierce
Engineering Science and
Mechanics
The Pennsylvania State
University
University Park, PA, USA
jtp5332@psu.edu

Glen Williams
Mechanical Engineering
The Pennsylvania State
University
University Park, PA, USA
gtw5020@psu.edu

Timothy W. Simpson
Mechanical Engineering
The Pennsylvania State
University
University Park, PA, USA
tws8@psu.edu

Nicholas A. Meisel
Engineering Design
The Pennsylvania State University
University Park, PA, USA
nam20@psu.edu

Christopher McComb
Engineering Design
The Pennsylvania State University
University Park, PA, USA
mccomb@psu.edu

ABSTRACT

Modern digital manufacturing processes, such as additive manufacturing, are cyber-physical in nature and utilize complex, process-specific simulations for both design and manufacturing. Although computational simulations can be used to optimize these complex processes, they can take hours or days--an unreasonable cost for engineering teams leveraging iterative design processes. Hence, more rapid computational methods are necessary in areas where computation time presents a limiting factor. When existing data from historical examples is plentiful and reliable, supervised machine learning can be used to create surrogate models that can be evaluated orders of magnitude more rapidly than comparable finite element approaches. However, for applications that necessitate computationally-intensive simulations, even generating the training data necessary to train a supervised machine learning model can pose a significant barrier. Unsupervised methods, such as physics-informed neural networks, offer a shortcut in cases where training data is scarce or prohibitive. These novel neural networks are trained without the use of potentially expensive labels. Instead, physical principles are encoded directly into the loss function. This method substantially reduces the time required to develop a training dataset, while still achieving the evaluation speed that is typical of supervised machine learning surrogate models. We propose a new method for stochastically training and testing a convolutional physics-informed neural network using the transient 3D heat equation- to model

temperature throughout a solid object over time. We demonstrate this approach by applying it to a transient thermal analysis model of the powder bed fusion manufacturing process.

1. INTRODUCTION

Cyber-physical manufacturing methods such as additive manufacturing (AM), can benefit in a variety of ways from the use of high-resolution, physics-based simulations [1]. For instance, these simulations are often essential for optimizing designs and fine-tuning process parameters and take advantage of the data-intensive nature of AM. For expensive manufacturing processes such as laser powder bed fusion (LPBF), simulations are especially important because they can reveal potential build failures or highlight areas where process parameters can be improved without wasting resources on failed build attempts [1].

While LPBF analysis can involve many different phenomena (e.g., thermomechanical, electricity and magnetism, dynamics), thermal analysis alone is particularly important because the powder fusion process is highly dependent on temperature. Accurately simulating LPBF thermal phenomena involves analyzing temperature changes at fast timescales below the nanosecond range [1], several orders of magnitude shorter than the entire build duration. As a result of this high temporal resolution, full simulations of LPBF can take hours or days [1]. In the context of the engineering design process, this time cost is often too large to be used to design parts with low production volumes or small profit margins. In order to decrease this

computational cost, some researchers have turned to creating surrogate models which can return approximate simulation results much more quickly than traditional simulations.

Artificial neural networks (ANNs), machine learning (ML) models that can be trained on existing simulation data, are a common means of constructing surrogate models. While there has been success in implementing computationally faster ANN surrogate models for LPBF, the training data is still generated from computationally-intensive methods such as finite element analysis (FEA) [2]. Generating training data from FEA simulations may take a long time, which makes developing and improving these models a very slow process. Thus, ANNs could be improved and iterated on rapidly if the need for computationally-intensive training datasets could be overcome. The ability to quickly tune LPBF models without first obtaining potentially expensive datasets would also reduce the logistical barrier behind starting the ML development process.

Physics-informed neural networks (PINN) are artificial neural networks supervised without labels. Instead, PINNs are constrained by partial differential equations (PDEs) [3]. By training an ANN using readily available analytical equations, the need for expensive training data can be eliminated while also encoding physical principles, which ensures that solutions are physically valid [4]. Since reducing development time of LPBF surrogate models is desirable, the ideal PINN should be designed with training data that is inexpensive to generate and utilize an architecture that has potential for universality [3]. Upon creating this PINN, the aggregate performance of this model must be compared to a traditional label-trained ANN that uses the same training data and architecture. In addition to this, an application for LPBF transient thermal analysis simulation is of great interest as well. Therefore, the research questions for this paper are:

1. How does a PINN perform compared to a traditional label-trained ANN over a large range of data in terms of aggregate prediction error?
2. How does the PINN compare, in terms of prediction error over time, to a traditional label-trained ANN based on a ground truth model in a LPBF thermal analysis problem?

2. BACKGROUND

This section provides relevant background information on LPBF (see Section 2.1), ML and ANN methods in LPBF (see Section 2.2) and related work involving PINNs (see Section 2.3).

2.1 An Overview of Laser Powder Bed Fusion Simulation

LPBF is a common metal additive manufacturing method used today. This process typically utilizes a radiative heat source that fuses powdered material together layer by layer. For each layer, a parametric laser path is generated and executed in order to fuse a particular shape. Although numerous variations of the LPBF process exist, ongoing LPBF research centers around incremental improvements in the areas of residual stress

reduction, surface finish optimization, melt pool optimization, build time reduction, part consistency, and failure rate reduction [1,5]. In addition to being experimentally investigated, these improvements are also driven by research involving LPBF simulations, which allow enhanced build attributes to be discovered without as much reliance on costly manufacturing trials [6].

A particularly active type of prediction in LPBF simulation research is of the melt pool produced by the heat source. Simulations can be used to estimate the melt pool shape, residual stresses, or thermomechanical dynamics [5]. This problem is particularly difficult to compute due to the disparity in time scales involved [1]. As models become more and more detailed, both the spatial and time scales become smaller. Since the overall build time for typical LPBF jobs can be hours or even days, every reduction in scale increases the difficulty of applying the model to entire builds, even with large CPU clusters [7].

2.2 Machine Learning in LPBF Systems

The applications of ML and ANNs in LPBF are varied. Of these applications, a substantial number of research efforts in this area involve process monitoring. The LPBF process is highly dependent on conditions within the build environment; so, gathering and processing data while LPBF builds are ongoing is a lucrative way of preventing failures or optimizing the process parameters. One particular application is using images of the melt pool in order to make predictions about the properties of the melt pool [8,9]. These images can be processed by an ANN in order to predict laser temperature, melt pool size, and defects. However, in order to make in situ adjustments to the LPBF process, potentially large amounts of data must be collected. Kwon et al. report several hundred gigabytes of image data for one build [9]. Yang et al. collected 118,928 images of melt pools after only 21 layers of an LPBF build [8].

It is difficult to assign labels to such a large number of images; therefore, an advantageous alternative is to apply physics domain knowledge to this problem. Ren et al. added physics-informed input features in addition to the melt pool images in order to help the training process [10]. For simulation purposes, ML methods have also been used to predict the outcome of the LPBF process; however, this method typically utilizes commercial simulation software to generate validation data [11]. In addition to the applications mentioned, an important concept to consider in the field of LPBF models is the ability to form hybrid models. Moges et al. demonstrated the success of a hybrid model involving the combination of traditional numerical analysis and ML/ANN-based surrogate models, improving computation speed and accuracy [12].

2.3 Related Work in Physics-Informed Neural Networks

PINNs are defined as any deep learning technique used to solve PDEs by encoding prior knowledge of the PDE in the loss function [3]. As such, there is a wide range of attempts to solve PDEs, or engineering problems involving PDEs, using this method. Raissi et al. introduced a framework for using PINNs to solve nonlinear PDEs for continuous and discrete time modes

[13]. This approach was further tested to converge to the PDE even with corrupted noise applied to the training dataset—suggesting that even with data that does not appear experimentally, a model can still converge [14]. This PINN framework by Raissi et al. also introduced the idea of using automatic differentiation of backpropagation instead of numerical differentiation for penalizing the physics-informed loss. In our work, we use numerical differentiation, but the method of automatic differentiation is common in most recent PINN research, with novel applications to Navier-Stokes equations, subsurface flow, noisy forward and inverse problems, time dependent PDEs, and performance iterations on the linear PDE PINN architecture [15–19]. In general, the architectures and training data used with PINNs is quite varied with each tailored to the specific problem being investigated.

For our work, one notable example is from Wang et al. who developed a non-automatic differentiation, finite time difference, PINN for transient electromagnetic analysis using a recurrent neural network (RNN) [20]. This approach was novel not only due to the use of an RNN for transient analysis, but also involved a unique way of generating training data, which is of great interest for our motivation. The problem space was a 2D uniformly spaced mesh, similar to how voxels are used in ANNs. Here, training data was created by randomly generating wave point sources and interference objects. The goal of this method was to quickly simulate abstract scenarios without actually collecting data or generating training data without using numerical methods—utilizing one of the main benefits of the PINN method towards inexpensive training. It is distinct from some of the automatic differentiation approaches that only lightly apply noise to training data instead of fully abstract data. Liu et al. created a physics constrained neural network (PCNN) for the 2D heat equation [21]. The PCNN method is similar to the PINN method in that it encodes physics domain knowledge; however, it still uses labelled training data in tandem with the physics-based loss function. The goal of this method is to reduce the amount of labelled data required, and to also enforce the physical constraint. For our work, the success of the heat equation PCNN is of particular interest for LPBF simulation.

3. PROPOSED METHOD AND ASSUMPTIONS

This section details the methods and assumptions regarding the LPBF finite difference method used here as a ground truth model (see Section 3.1), the layer architecture shared by the PINN and the label-trained ANN (see Section 3.2), the LPBF application experimental design (see Section 3.3), and the experimental approach used to compare the PINN and label-trained ANN (see Section 3.4).

3.1 Finite Difference Method for LPBF Ground Truth

For this paper, we build on the 3D finite difference (FD) method tailored for AM proposed by Stockman et al. [22]. This model analyzes the transient temperature of a volume of fused powder in a laser direct energy deposition (DED) machine and includes conduction, a radiative boundary, a convective boundary, and Dirichlet boundary conditions. DED is similar to

LPBF in that it often uses a laser to fuse powdered metal into a specific shape; however, it does not use a large bed of powder.

To represent the LPBF process instead, internal heat generation (energy transfer) was added in place of the constant temperature Dirichlet boundary condition used to approximate the laser source. Approximating the laser source as internal heat generation is commonly used in FEA LPBF simulation [23,24]. There are many approaches to modeling this internal heat generation, but for our approach, one voxel was heated at a time [25]. The radiative boundary was not included. For voxels that only pertain to conduction and Dirichlet boundary conditions, the equation is

$$T_{i,j,k}^{n+1} = Fo(T_{i+1,j,k}^n + T_{i-1,j,k}^n + T_{i,j+1,k}^n + T_{i,j-1,k}^n + T_{i,j,k+1}^n + T_{i,j,k-1}^n + \frac{Q_{i,j,k}^n}{k_p} - 6T_{i,j,k}^n) + T_{i,j,k}^n \quad (1)$$

where $T_{i,j,k}^n$ is the absolute temperature at the current timestep (n) of a node at positions i, j, k (corresponding to the x, y, z coordinates), $T_{i,j,k}^{n+1}$ is absolute temperature at the future ($n+1$) timestep, $Q_{i,j,k}^n$ is current timestep (n) heat generated at node i, j, k , and Fo is the Fourier number found in Equation 2

$$Fo = \frac{k_p dt}{h^2 \rho c} \quad (2)$$

where k_p is the thermal conductivity of the powder, dt is the timestep, h is the cubic voxel side length, ρ is the density of the powder, and c is the specific heat of the powder. For voxels that involve conduction and convection the equation is modified as

$$T_{i,j,k}^{n+1} = Fo[(2 Bi T_{i,j-1,k}^n) + (2 T_{i,j+1,k}^n) + T_{i,j,k+1}^n + T_{i+1,j,k}^n + T_{i-1,j,k}^n + T_{i,j,k-1}^n + T_{i,j,k}^n + \frac{Q_{i,j,k}^n}{k_p} + T_{i,j,k}^n (-2 Bi - 6)] + T_{i,j,k}^n \quad (3)$$

where Bi is the Biot number computed as

$$Bi = \frac{C * h}{k_a} \quad (4)$$

where C is the heat transfer coefficient of the air and k_a is the thermal conductivity of the air.

By using the finite difference method in conjunction with voxel representation, the equation that calculates temperature at each voxel can be generalized for every volumetric region of interest in the powder volume to a desired resolution. Equation 1 applies to all interior voxels connected to the conduction voxels seen in Figure 1 and all other interior voxels not connected to a boundary condition. Meanwhile, Equation 3 applies to all interior voxels connected to convection voxels (see Figure 1).

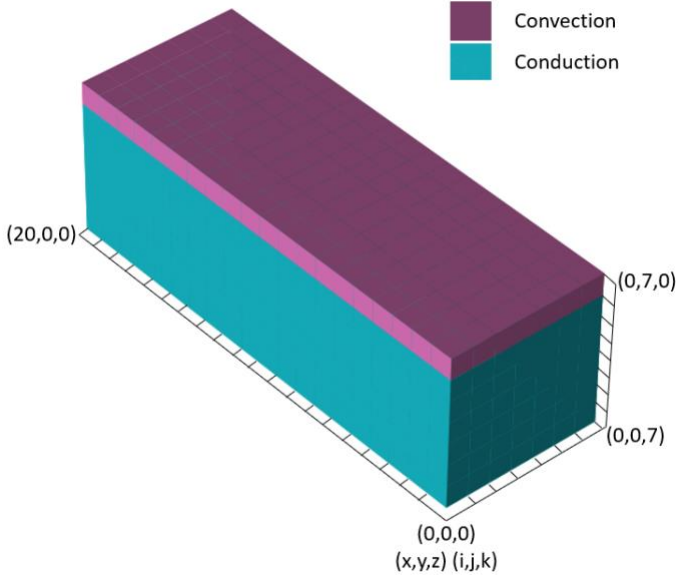


FIGURE 1: Visualization of boundary condition locations used with the finite difference equations 1 and 3.

3.2 ANN Architecture Design

In order to facilitate comparison between the PINN and the label-trained ANN, the training parameters, training data, and physical constants of the PINN and the label-trained ANN were held constant. This section details the shared ANN architecture of the PINN and the label-trained ANN (see Section 3.2.1), custom loss function of the PINN (see Section 3.2.2), and design of the shared training dataset (see Section 3.2.3).

3.2.1 PINN and label-trained ANN architecture

The simulated powder bed volume used as the input for both the PINN and the label-trained ANN had a voxel dimension of $20 \times 7 \times 7$. Each of the voxels encoded a two-channel vector representing the temperature and heat generation magnitude at each voxel at a particular instance in time. The output of the PINN and the label-trained ANN was a $20 \times 7 \times 7$ voxel-based field, where each voxel represents the estimated temperature of the region after one additional timestep has passed. Both of these models are discrete time models, where input data describing the temperature and heat transfer field is used to predict a temperature field at a fixed timestep. The loss function of the PINN only iterates through the interior nodes of the input since the outer layers of the input space are encoded as the boundary conditions for the temperature channel (see Figure 1) or exist as “padding” for the heat generation channel. In order to match the dimensionality of the input, the outer voxels of the output also serve as padding.

Since the LPBF application requires prediction of an entire scalar field throughout a volume, it necessitates a model with a relatively high output dimensionality. A convolutional neural network (CNN) architecture was selected because CNNs are particularly advantageous at retaining practicable training times with increases in input and output dimensionality [26]. The

shared architecture of both the PINN and label-trained ANN is specified in Figure 2 for our work.

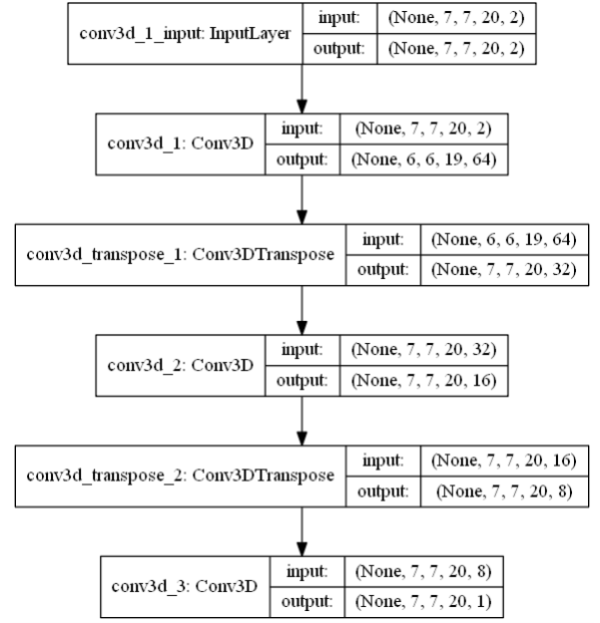


FIGURE 2: Flowchart of shared PINN and label-trained ANN architecture layers

The shared architecture in Figure 2 was developed in TensorFlow version 2.1.0, Keras version 2.3.1, and Python version 3.7.6. The ANNs were trained using a GTX 1080 Ti GPU while the FD was executed using an intel 4770k CPU. The PINN and the label-trained ANN were trained using the same 5000 training samples with a batch size of 100 samples for 50 epochs. The optimizer used was Adam with a learning rate of 5×10^{-4} . The label-trained ANN used mean squared error (MSE) as the loss function while the PINN used the loss function described in Section 3.2.2.

3.2.2 PINN loss function

The PINN custom loss function is constructed as the MSE of the conduction and convection finite difference equations. Since loss should approach 0 during training, the future time step solution is subtracted from the current temperature and FD calculated change in temperature. The loss equation corresponding to voxels with only conduction and heat generation is given by

$$Loss_a = \sum_{i=1}^7 \sum_{j=2}^7 \sum_{k=1}^{20} |T_{i,j,k}^n + (\Delta T_{i,j,k})_{cond} - T_{i,j,k}^{n+1}| \quad (5)$$

where $T_{i,j,k}^n$ is the current (n) timestep temperature of the voxel at position i, j, k (indices corresponding to the x, y , and z axes), $(\Delta T_{i,j,k})_{cond}$ is the change in temperature calculated using equation 1, and $T_{i,j,k}^{n+1}$ is the future ($n + 1$) temperature predicted by the PINN. The loss equation corresponding to voxels with conduction, convection, and heat generation is computed as

$$Loss_b = \sum_{i=1}^7 \sum_{j=1}^1 \sum_{k=1}^{20} |T_{i,j,k}^n + (\Delta T_{i,j,k}^n)_{conv} - T_{i,j,k}^{n+1}| \quad (6)$$

where $T_{i,j,k}^n$ is the current (n) timestep temperature of the voxel at position i, j, k (again corresponding to the x, y , and z axes), $(\Delta T_{i,j,k}^n)_{conv}$ is the change in temperature calculated using equation 3, and $T_{i,j,k}^{n+1}$ is the future ($n + 1$) temperature predicted by the PINN. The loss equations are then combined using

$$Loss = \frac{(Loss_a + Loss_b)^2}{N} \quad (7)$$

where N is the number of voxels included in the loss calculation (here, $N = 450$).

3.2.3 Training dataset design

In an effort to eliminate the need for prohibitive labelled data, synthetic data can be particularly advantageous. Existing PINN methods incorporate many different approaches for generating training data without relying on costly or unavailable experimental observations. For instance, one approach used experimental data with 1% noise [14]. A second approach randomly generated shapes in order to mimic experimental training data [20].

In this work, we sought a method of training set generation that maximized the speed of generating training data while minimizing the potential bias toward a small number of researcher-selected shapes. To achieve this goal, we procedurally generated a set of temperature and heat generation fields in which each voxel was uniformly assigned a random value between -100 and 100 (for both temperature, measuring in K, and energy transfer, measured in W). This range was intentionally below the order of magnitude of temperatures and approximately on the order of magnitude for heat generation values experienced in a powder bed volume after melt pool solidification in order to keep the input data uniform and simple. Here, positive values represent heat generation within a voxel, and negative values represent heat absorption. For temperature, although half of the data is below 0 K, the loss function used is continuous for all temperature values. In other application areas, the selection of the range for the random values should be treated as an additional hyperparameter in the end-to-end PINN training process. An example of a training data sample (for the temperature or heat transfer channel) can be seen in Figure 3.

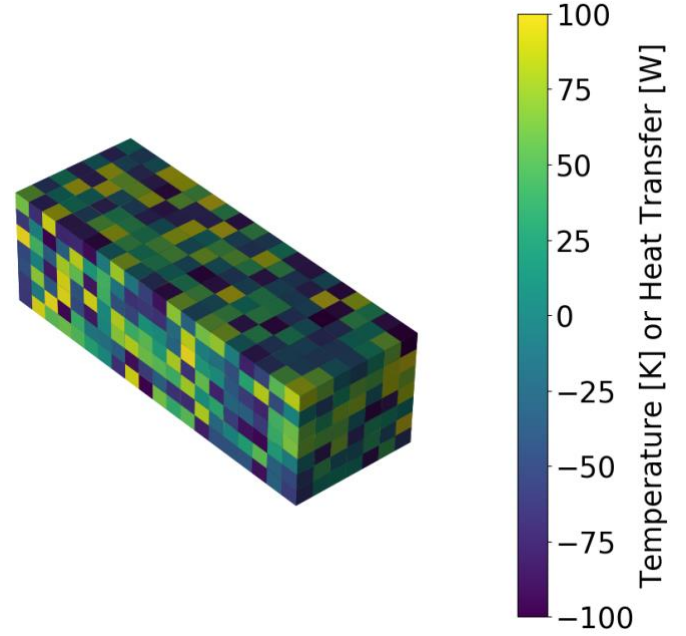


FIGURE 3: Example 3D plot of uniformly randomly generated temperature channel and heat transfer channel training data.

In addition to the training data, the training process is constrained by global constants. The temperature boundary conditions are held constant at a value of 293 K. The cubic voxel side length was 1×10^{-7} meters with a constant timestep length of 2.7×10^{-8} seconds. This voxel size is below the typical size for powdered metal in LPBF; however, this model only simulates an effective homogeneous medium [27]. The material properties chosen were: 0.1 W/m·K for the powder thermal conductivity, 0.02 W/m·K for the air thermal conductivity, 25 watts per meters squared kelvin for the heat transfer coefficient, 8000 kg/m³ for the powder density, and 420 J/K·kg [27–30]. These values are within the range of numbers found in metal LPBF processes. An important constant to mention is the thermal conductivity of the powder. While metal powders typically used in LPBF have conductivities of 10–200 W/m·K, research shows that due to the packing density and composition of the air, the effective value is much lower [30].

3.3 LPBF Application Problem Statement

The performance of the PINN was assessed in a transient thermal analysis problem of a LPBF process. As previously mentioned, the finite difference method used for the loss function is already tailored for additive manufacturing [22], with some modification as noted in Section 3.1. Heat conduction is modeled with Dirichlet boundary conditions on 5 exterior faces of the temperature field, and a convection boundary condition on the 6th surface. The laser is modeled using a single internal heat source approach, where 1 voxel is heated at a time. The boundary conditions on all 6 faces are constrained by a temperature value of 293 kelvin.

The simulation involves two sequential modes: a single laser track and a diffusion mode with no heat generation. From $t = 0$

to 504 nanoseconds- a 1000-watt heat source moves voxel by voxel along the top plane of the temperature field. This approximates a laser moving at 3.57 m/s. These numbers are within the range of values found in LPBF processes [31,32]. After the $t = 504$ nanoseconds, the heat source is turned off, and the temperature field approaches steady state until $t = 2772$ nanoseconds is reached. A cutaway of the FD simulation result at $t = 504$ nanoseconds is shown in Figure 4.

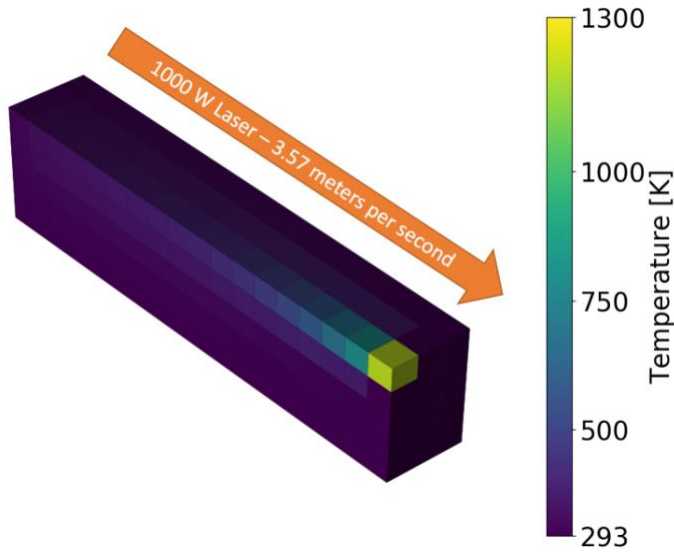


FIGURE 4: Visualization showing cutaway view of the ground truth temperature through the powder bed volume after 504 nanoseconds of laser radiation.

3.4 Experimental Methods

To investigate differences in the PINN training method and the label-trained method, we compared the aggregate prediction error of 30 pairs of the PINN and label-trained ANN in a one-sided Wilcoxon signed-rank test. Pairs of models were trained using the same random seed. Each pair of PINN and label-trained ANN models was trained with the parameters found in Section 3.2.3 with a different random number seed for each pair. Each pair of PINN and label-trained ANN models was also tested on a dataset generated similarly to the training dataset. However, instead of values between -100 and 100, we selected 1000 samples of a much larger range of 100-10000 Kelvin and 100-10000 Watts in order to test unseen data out of the training range and demonstrate universality of the model. Each pair also used a shared seed for a total of 30 training datasets, 30 testing datasets, and 30 different seed pairs. Each sample is only fed through the network once; so, this does not evaluate compounding error.

We tested each PINN and label-trained ANN model by directly comparing the predictions to the FD ground truth (ground truth minus prediction) and calculating the mean of these prediction errors per model. These means were then compared using a Wilcoxon signed-rank test [33] in order to demonstrate the difference between the two different training methods aggregate error biases. In order to compare the distribution of prediction error between the two models, the mean standard deviation of the 30 pairs was also calculated.

The FD LPBF simulation serves as the ground-truth model for the PINN method since the PINN loss function is derived from the same FD equations. For demonstrative purposes, we selected the two best PINN and label-trained ANN models from the 30 models tested in the Wilcoxon signed-rank test (based on mean error).

The FD LPBF simulation is compared to the ANNs in two ways: (1) mean absolute percent error (MAPE) over time and (2) a visual inspection of error per voxel over time. MAPE was chosen as a way of continuously evaluating the aggregate temperature field error of each timestep. By evaluating this metric over time, we can learn how each ANN performs in each simulation mode defined in Section 3.3. By calculating and displaying the error for each voxel over time, we can also empirically evaluate the source of error for each ANN.

4. RESULTS

This section presents the results of the Wilcoxon signed-rank test (see Section 4.1) and LPBF simulation (see Section 4.2) based on the methods previously described (see Section 3.4). These results test our two research questions in Section 1.

4.1 Comparison of PINN and Label-Trained ANN

For the first research question, we assess how PINN models compare to label-trained ANN models. Training results of one sample of seed-paired PINN and label-trained ANN models are shown in Figure 5. Both training methods converged similarly, but to different loss values due to the different loss functions used.

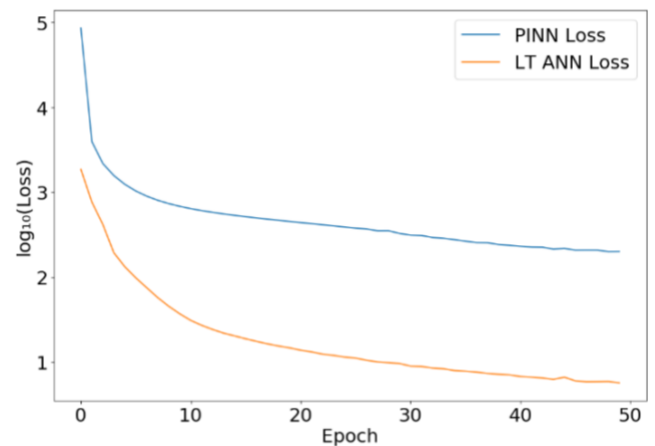


FIGURE 5: Plot of example loss per epoch convergence for PINN and label-trained ANN models

The Wilcoxon signed-rank test dataset was calculated for the aggregate prediction error of 30 pairs of PINN and label-trained models. This test gave a p-value of 0.145, showing with moderate significance that the PINN has a different aggregate prediction bias compared to the label-trained ANN. In addition to the statistical test, we also show the distribution of predictions from the two best models (based on mean bias) in Figure 6.

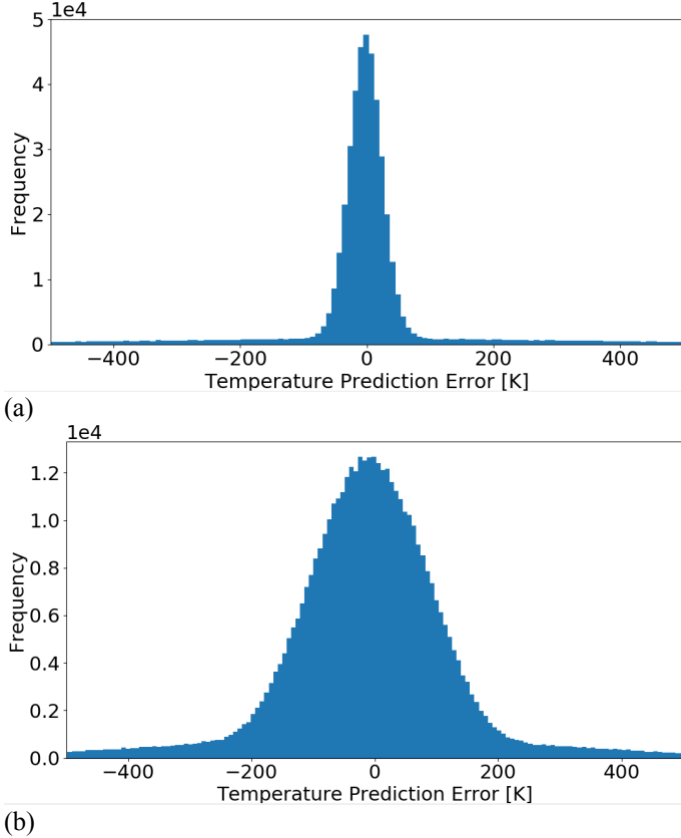


FIGURE 6: Example histograms of the two best (a) PINN and (b) label-trained ANN predicted temperatures vs. the FD ground truth temperature error residuals

Although in general the PINN predictions appear to have a smaller standard deviation, the actual mean standard deviation across all 30 pairs is 150.0 for the PINN and 149.8 for the label-trained ANN. Because the standard deviations are relatively similar, the absolute value of the aggregate prediction bias acts as a better predictor of model performance.

4.2 LPBF Inspired Simulation

For the second research question, we assess how the PINN performs over time in a LPBF simulation compared to the label-trained ANN. The LPBF simulation involves compounding time steps and as a result produced more divergent results between the PINN and label-trained ANN than the previous section, even though both models selected had relatively low mean biases. For the PINN, MAPE peaked at a value of 1.24% while the label-trained ANN peaked at a value of 22.5%. In the laser track simulation mode, both models increased in error over time. In the conduction simulation mode, the PINN system approaches 0 error while the label-trained ANN does not. These results can be found in Figure 7.

To visualize the transient simulation process, eight instances of the LPBF simulation are shown for comparison in Figure 8. The first four instances ($t = [0, 504]$ nanoseconds) demonstrate the laser track motion; the last four ($t = (504, 2772]$ nanoseconds) demonstrate the diffusion mode with no heat generation. In the

first three rows, we plot the PINN, FD ground truth, and label-trained ANN simulations at the chosen timesteps. In the last two rows, we plot the error of the PINN and label-trained ANN predictions vs the ground truth FD model.

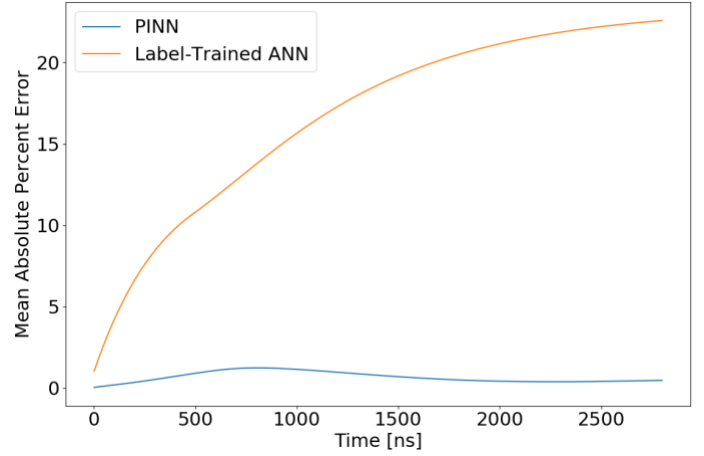


FIGURE 7: Plot of MAPE vs. time for the PINN and label-trained ANN LPBF simulation

Visually, the simulations appear very similar to the FD ground-truth; however, the pattern of the error is dissimilar for the PINN and label-trained ANN. For the first four instances (the laser track simulation mode), the PINN error mirrors the laser track. In the last four instances (the diffusion simulation mode), the error is more symmetric throughout the temperature field and decreases over time. The label-trained ANN does not exhibit the same pattern of error as the PINN, it is more symmetric for all eight timesteps- with error concentrated towards the center of the temperature field rather than the boundary. The error for the label-trained ANN also increases over time.

5. CONCLUSION

In the case of LPBF, a cyber-physical AM method, computational simulations become increasingly important towards optimizing the fabrication process and preventing failure. In contribution to this paradigm, ANNs and ML have become powerful tools for LPBF. In this work, the goal was to evaluate a method to eliminate the prohibitive data labelling process and to train surrogate models using well understood physics domain knowledge. Our work created a LPBF heat transfer simulation that utilized a stochastically-trained PINN with multiple boundary conditions.

For the first research question, we investigated the statistical difference between our proposed PINN and a traditionally label-trained ANN. Our Wilcoxon signed-rank test demonstrated that, in general, it is likely that the PINN performs better than the label-trained ANN given the same training data and constraints. However, although the distribution of prediction errors appeared to be superior for the PINN, the standard deviations of both the PINN and label-trained ANNs prediction errors were the same. Knowing this, at a very minimum the PINN performs equally to the label-trained ANN.

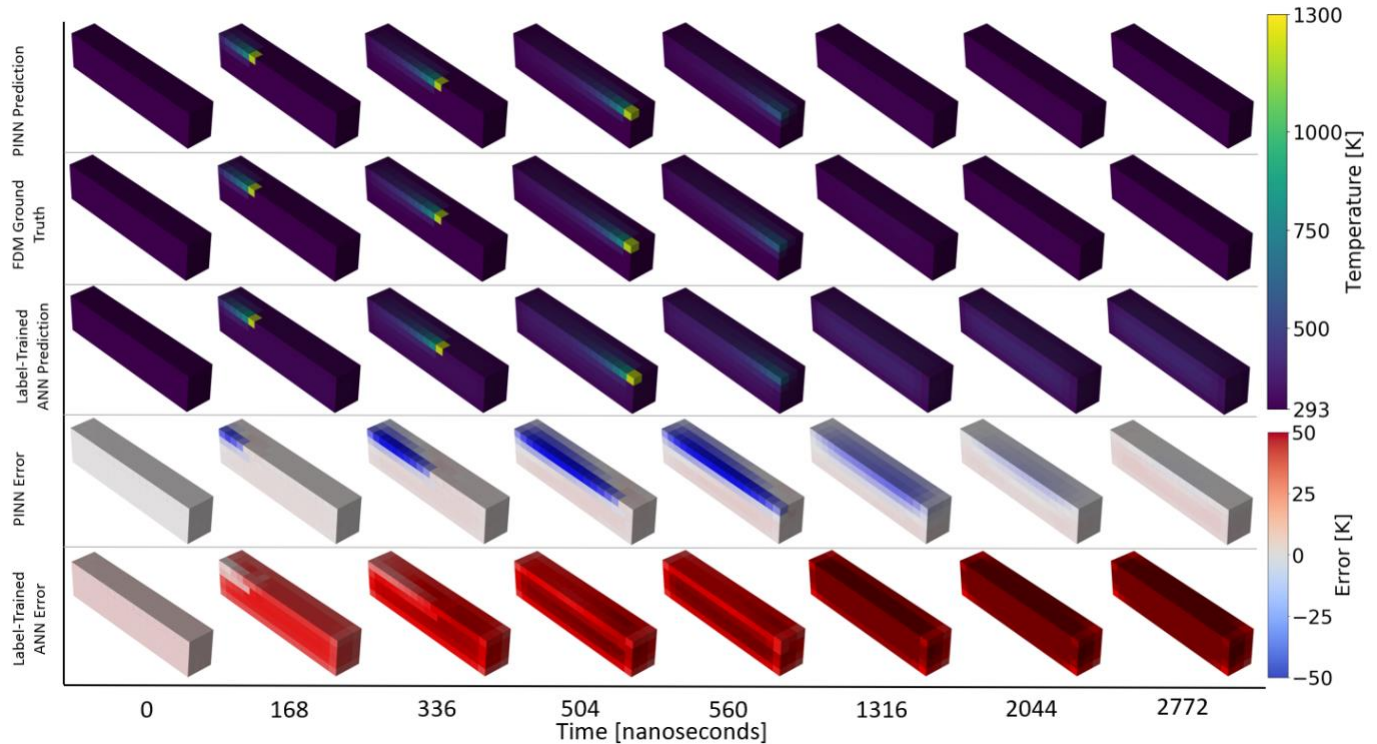


FIGURE 8: Cutaway plots of PINN simulation, FD ground truth simulation, label-trained ANN simulation, PINN error, and label-trained ANN error at selected timesteps.

The second research question involved simulating a LPBF thermal problem. For this problem, the error was compounded over time, which further demonstrated the difference between the PINN and label-trained models. Upon comparing the two best models from the statistical test dataset, the PINN showed better MAPE over time. Furthermore, upon visual analysis of the errors for each voxel, it was clear that the pattern of error was different for the PINN compared to the label-trained ANN.

Although our proposed PINN showed promise compared to the traditionally label-trained ANN, especially in the LPBF simulation, this work only involved a single PDE, namely, the heat equation. This work also used many static physical constants which can change over time in a LPBF build. Future work can directly build on this work by investigating the ability of the PINN to learn more variables or channels at once (beyond the temperature and heat generation channels demonstrated here). The LPBF process involves many other physical phenomena, and future work could investigate similar simulations with different loss functions. In addition to using different PDEs for loss functions, the method of an auto-differentiating PINN could be investigated for LPBF as well.

ACKNOWLEDGEMENTS

This research is based upon work supported by the National Science Foundation (Grant No. CMMI-1825535). 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] King, W. E., Anderson, A. T., Ferencz, R. M., Hodge, N. E., Kamath, C., Khairallah, S. A., and Rubenchik, A. M., 2015, "Laser Powder Bed Fusion Additive Manufacturing of Metals; Physics, Computational, and Materials Challenges," *Appl. Phys. Rev.*, **2**(4), p. 041304.
- [2] Stathatos, E., and Vosniakos, G.-C., 2019, "Real-Time Simulation for Long Paths in Laser-Based Additive Manufacturing: A Machine Learning Approach," *Int. J. Adv. Manuf. Technol.*, **104**(5–8), pp. 1967–1984.
- [3] Shin, Y., 2020, "On the Convergence of Physics Informed Neural Networks for Linear Second-Order Elliptic and Parabolic Type PDEs," *Commun. Comput. Phys.*, **28**(5), pp. 2042–2074.
- [4] Elhamod, M., Bu, J., Singh, C., Redell, M., Ghosh, A., Podolskiy, V., Lee, W.-C., and Karpadne, A., 2020, "CoPhy-PGNN: Learning Physics-Guided Neural Networks With Competing Loss Functions for Solving Eigenvalue Problems," 31st AAAI Conf. Artif. Intell. AAAI 2017, **1**(1), pp. 2576–2582.
- [5] Ganeriwala, R. K., Hodge, N. E., and Solberg, J. M., 2021, "Towards Improved Speed and Accuracy of Laser Powder Bed Fusion Simulations via Multiscale Spatial Representations," *Comput. Mater. Sci.*, **187**(June 2020), p. 110112.
- [6] Megahed, M., Mindt, H. W., Willems, J., Dionne, P., Jacquemetton, L., Craig, J., Ranade, P., and Peralta, A., 2019, "LPBF Right the First Time—the Right Mix

- Between Modeling and Experiments,” *Integr. Mater. Manuf. Innov.*, **8**(2), pp. 194–216.
- [7] Koepf, J. A., Gotterbarm, M. R., Markl, M., and Körner, C., 2018, “3D Multi-Layer Grain Structure Simulation of Powder Bed Fusion Additive Manufacturing,” *Acta Mater.*, **152**, pp. 119–126.
- [8] Yang, Z., Lu, Y., Yeung, H., and Kirshnamurthy, S., 2020, “3D Build Melt Pool Predictive Modeling for Powder Bed Fusion Additive Manufacturing,” *Volume 9: 40th Computers and Information in Engineering Conference (CIE)*, American Society of Mechanical Engineers.
- [9] Kwon, O., Kim, H. G., Kim, W., Kim, G.-H., and Kim, K., 2020, “A Convolutional Neural Network for Prediction of Laser Power Using Melt-Pool Images in Laser Powder Bed Fusion,” *IEEE Access*, **8**, pp. 23255–23263.
- [10] Ren, Y., and Wang, Q., 2020, “Physics-Informed Gaussian Process Based Optimal Control of Laser Powder Bed Fusion,” *Volume 2: Intelligent Transportation/Vehicles; Manufacturing; Mechatronics; Engine/After-Treatment Systems; Soft Actuators/Manipulators; Modeling/Validation; Motion/Vibration Control Applications; Multi-Agent/Networked Systems; Path Planning/Motion Control*, American Society of Mechanical Engineers, pp. 1–8.
- [11] Ren, Y., Wang, Q., and Michaleris, P., 2019, “Machine-Learning Based Thermal-Geometric Predictive Modeling of Laser Powder Bed Fusion Additive Manufacturing,” *ASME 2019 Dyn. Syst. Control Conf. DSCC 2019*, **1**, pp. 1–10.
- [12] Moges, T., Yang, Z., Jones, K., Feng, S., Witherell, P., and Lu, Y., 2021, “Hybrid Modeling Approach for Melt Pool Prediction in Laser Powder Bed Fusion Additive Manufacturing,” *J. Comput. Inf. Sci. Eng.*, pp. 1–46.
- [13] Raissi, M., Perdikaris, P., and Karniadakis, G. E., 2017, “Physics Informed Deep Learning (Part I): Data-Driven Solutions of Nonlinear Partial Differential Equations,” (Part I), pp. 1–22.
- [14] Silvestri, M., Lombardi, M., and Milano, M., 2020, “Injecting Domain Knowledge in Neural Networks: A Controlled Experiment on a Constrained Problem,” (Part II), pp. 1–19.
- [15] Jin, X., Cai, S., Li, H., and Karniadakis, G. E., 2021, “NSFnets (Navier-Stokes Flow Nets): Physics-Informed Neural Networks for the Incompressible Navier-Stokes Equations,” *J. Comput. Phys.*, **426**(Hui Li).
- [16] Jin, X., Cai, S., Li, H., and Karniadakis, G. E., 2020, “NSFnets (Navier-Stokes Flow Nets): Physics-Informed Neural Networks for the Incompressible Navier-Stokes Equations,” *Water Resour. Res.*, **56**(5), pp. 1–16.
- [17] Yang, Y., and Perdikaris, P., 2019, “Adversarial Uncertainty Quantification in Physics-Informed Neural Networks,” *J. Comput. Phys.*, **394**, pp. 136–152.
- [18] Meng, X., Li, Z., Zhang, D., and Karniadakis, G. E., 2019, “PPINN: Parareal Physics-Informed Neural Network for Time-Dependent PDEs,” *Comput. Methods Appl. Mech. Eng.*, **370**, p. 113250.
- [19] Kharazmi, E., Zhang, Z., and Karniadakis, G. E. M., 2019, “Variational Physics-Informed Neural Networks For Solving Partial Differential Equations,” *arXiv*, pp. 1–24.
- [20] Noakoasteen, O., Wang, S., Peng, Z., and Christodoulou, C., 2020, “Physics-Informed Deep Neural Networks for Transient Electromagnetic Analysis,” *IEEE Open J. Antennas Propag.*, **1**(August), pp. 404–412.
- [21] Liu, D., and Wang, Y., 2019, “Multi-Fidelity Physics-Constrained Neural Network and Its Application in Materials Modeling,” *Volume 2A: 45th Design Automation Conference*, American Society of Mechanical Engineers, pp. 1–12.
- [22] Stockman, T., Schneider, J. A., Walker, B., and Carpenter, J. S., 2019, “A 3D Finite Difference Thermal Model Tailored for Additive Manufacturing,” *JOM*, **71**(3), pp. 1117–1126.
- [23] Shen, N., and Chou, K., 2012, “Thermal Modeling of Electron Beam Additive Manufacturing Process: Powder Sintering Effects,” *ASME 2012 International Manufacturing Science and Engineering Conference*, American Society of Mechanical Engineers, pp. 287–295.
- [24] Roberts, I. A., Wang, C. J., Esterlein, R., Stanford, M., and Mynors, D. J., 2009, “A Three-Dimensional Finite Element Analysis of the Temperature Field during Laser Melting of Metal Powders in Additive Layer Manufacturing,” *Int. J. Mach. Tools Manuf.*, **49**(12–13), pp. 916–923.
- [25] Zhang, Z., Huang, Y., Rani Kasinathan, A., Imani Shahabad, S., Ali, U., Mahmoodkhani, Y., and Toyserkani, E., 2019, “3-Dimensional Heat Transfer Modeling for Laser Powder-Bed Fusion Additive Manufacturing with Volumetric Heat Sources Based on Varied Thermal Conductivity and Absorptivity,” *Opt. Laser Technol.*, **109**(May 2018), pp. 297–312.
- [26] Tan, M., and Le, Q. V., 2019, “EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks,” *36th Int. Conf. Mach. Learn. ICML 2019*, **2019-June**, pp. 10691–10700.
- [27] Slotwinski, J. A., Garboczi, E. J., Stutzman, P. E., Ferraris, C. F., Watson, S. S., and Peltz, M. A., 2014, “Characterization of Metal Powders Used for Additive Manufacturing,” *J. Res. Natl. Inst. Stand. Technol.*, **119**, p. 460.
- [28] Masoomi, M., Soltani-Tehrani, A., Shamsaei, N., and Thompson, S. M., 2020, “Convection Heat Transfer Coefficients for Laser Powder Bed Fusion,” *Solid Free. Fabr. 2018 Proc. 29th Annu. Int. Solid Free. Fabr. Symp. - An Addit. Manuf. Conf. SFF 2018*, pp. 1686–1693.
- [29] Ripetskiy, A., Zelenov, S., Kuznetsova, E., and Rabinskiy, L., 2018, “Evaluation of the Thermal Processes and Simulation Methods for Additive Manufacturing Based on the Geometry Voxel

- Representation,” *Key Eng. Mater.*, **771**, pp. 91–96.
- [30] Ladani, L., Romano, J., Brindley, W., and Burlatsky, S., 2017, “Effective Liquid Conductivity for Improved Simulation of Thermal Transport in Laser Beam Melting Powder Bed Technology,” *Addit. Manuf.*, **14**, pp. 13–23.
 - [31] Buchmayr, B., Panzl, G., Walzl, A., and Wallis, C., 2017, “Laser Powder Bed Fusion - Materials Issues and Optimized Processing Parameters for Tool Steels, AlSiMg- and CuCrZr-Alloys,” *Adv. Eng. Mater.*, **19**(4), p. 1600667.
 - [32] Ninpetch, P., Kowitwarangkul, P., Mahathanabodee, S., Tongsri, R., and Ratanadecho, P., 2019, “Thermal and Melting Track Simulations of Laser Powder Bed Fusion (L-PBF),” *IOP Conf. Ser. Mater. Sci. Eng.*, **526**(1), p. 012030.
 - [33] Durango, A., and Refugio, C., 2018, “An Empirical Study on Wilcoxon Signed Rank Test,” *J. Negros Orient. State Univ.*, (December).