

Application of Scientific Machine Learning Techniques to Metal Additive Manufacturing Simulation: A Comparison Between Physics-Informed Neural Networks and Extended Physics-Informed Neural Networks

Bohan Peng* and Ajit Panesar†
Imperial College London, London, England, SW7 2AZ, United Kingdom

This paper presents a comparison between physics-informed neural network (PINN) and extended physics-informed neural network (XPINN), both scientific machine learning techniques, when applied to the same multi-layer simulation framework for multi-layer Direct Energy Deposition (DED) thermal history prediction. Whilst both techniques offer the benefits of being meshless and having derivative information readily available, they each have their own caveats. XPINN's 'divide-and-conquer' strategy overcomes the classic deficiency of PINN when working with discontinuities at an increased computational cost. This comparison study aims to identify the threshold in complexity beyond which the increased computational cost is justified by the enhanced expressibility of XPINN.

I. Introduction

Additive manufacturing (AM), a key pillar of Industry 4.0 [1], is playing an increasingly critical role in realising complex designs during actual production in many industries (e.g. aerospace, biomedical, high-end automobile etc.). The increased adoption in production implies an increasing demand for first-time-right from AM, especially the metal AM (MAM) processes due to the high material cost and the usually high-value parts. However, achieving first-time-right remains a challenge due to the complexity and extreme conditions during the procedure. During the common MAM processes, the feedstock (in Direct Energy Deposition (DED)) or the layer of powder (in Laser Powder Bed Fusion (L-PBF) or Electron Beam Melting (EBM)) is melted with a local heat source (commonly laser and electron beam) and re-solidifies to form the designed part geometry. Such a process involves steep thermal gradients and high cooling rates, resulting in multi-scale impacts on the final product – from impacts on mechanical properties due to effects on microstructure to impacts on the final geometry due to thermally induced part distortion [2, 3]. It consequently leads to complications in part qualification and, in the worst case, part failure during printing. Hence, there is no shortage of attempts to predict the thermally induced phenomena in the literature with a significant amount of publications utilising the numerical approach [4]. The high-fidelity attempts with explicit scan pattern modelling are usually restricted to a limited number of layers [5]. By neglecting certain thermal phenomena such as phase transformation, Parry et al. [6] and Cheng et al. [7] were able to investigate the impact of scan patterns relatively quickly with their own set of assumptions on the heat source and/or material model, respectively. Another popular, fast approach, established and matured by Prof. Albert To's group, is based on the modified inherent strain (MIS) method where a detailed simulation over a small scale is used to calibrate the inherent strain which is subsequently used to predict the overall distortion at part level [8–12]. The recent progress by the same group, however, re-introduces thermal simulation since the simple calibration results in larger discrepancy when large DED part is considered [12]. The classic compromise between fidelity and computational time persists in the numerical solution methods.

The rapid progress in machine learning (ML) algorithms and the enabling hardware over the recent decade have led to a significant uptake of ML-based surrogate modelling in many traditionally computationally intensive areas. There is

*Research Postgraduate, IDEA Lab, Department of Aeronautics

† Associate Professor (Reader), IDEA Lab, Department of Aeronautics. Corresponding author: a.panesar@imperial.ac.uk

no exception in the field of MAM. In Demir et al.'s work, a four-layer neural network (NN) is trained on simulation data and used to predict the residual stresses based on the scan path [13]. Also related to scan path, Ren et al. proposed an architecture which blends a recurrent neural network (RNN) consisting of long short-term memory cells and a NN of one hidden layer in order to predict the thermal history based on the scan path. The model is trained on simulation data based on FEA [14]. Another use of NN is to generate compensation for overhang angle to improve the print accuracy of lattice structures through MAM [15]. The convolutional neural network (CNN) is another popular architecture used in AM-related works for its better graphics-based learning capability. For example, a CNN can be trained with CFD-based, mesoscale thermal simulation data to predict the melt pool depth given a surface temperature during the L-PBF process [16]. With most ML-based methods for AM-related applications being purely data-driven, there is usually a common question on their applicability and a stigma of being 'black-box'. It leads to the concept of scientific machine learning (SciML) which aims to incorporate physical laws into the training process of the ML models. Physics-informed Neural Network (PINN) [17] is one of the most widely adopted SciML methods since its conception in 2019. In the field of MAM, there have also been some attempts to employ PINN for thermally-related simulations. Liao et al. demonstrated PINN's capability of solving both the forward problem of obtaining the temperature history of one scan in DED and the inverse problem of obtaining material thermal properties from synthesised data [18]. In [19], PINN is used to predict the temperature and the melt pool characteristics during L-PBF with only a moderate amount of data for training. Strategies have been proposed and implemented in [20], enabling PINN to overcome the difficulty in capturing the discontinuity in the initial condition for multi-layer DED thermal history prediction without adding observation data. On the theoretical front to address the general deficiency of PINN in accounting for discontinuities, extended-PINN (XPINN) has also been proposed in [21]. There has also been analysis on the occasions where XPINN improves on PINN's performance [22].

In this article, a SciML-based, multi-layer framework for DED thermal history simulation with either PINN or XPINN as the backend is proposed, allowing part-scale, meshless thermal history prediction. The PINN-based implementation applies some of the strategies presented in [20]. It is compared against the XPINN-based implementation in two case studies of configurations with an increasing level of complexity.

II. Methodology

To simplify the explanation, Figure 1 illustrates the four main zones considered during the DED simulation: 1) substrate; 2) Zone A: already-printed layer; 3) Zone B: region in the current layer where the material is deposited; 4) Zone C: region in the current layer where the material is not yet deposited.

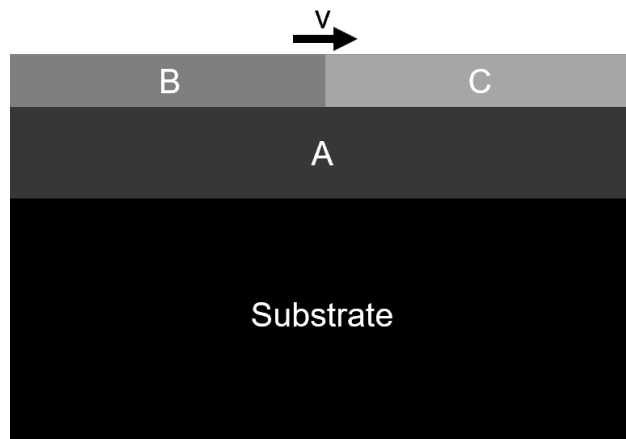


Fig. 1 Key zones during the multi-layer process: Zone A, B, and C constitute the part region. Zone A is the already printed layer. Zone B and C constitute the layer that is being printed – Zone B has material deposited and Zone C is where the material will be deposited as indicated by the velocity vector (v) of the heat source.

PINN, a type of SciML-based method, is used as a forward solver for the temperature history during the DED process and Figure 2 illustrates the architecture of the NN employed. At the core of PINN, the auto-differentiation [23] capability of an NN is utilised to obtain the respective terms in the governing equations. It subsequently allows the ‘physics-informed’ loss terms to be computed and used in the training process.

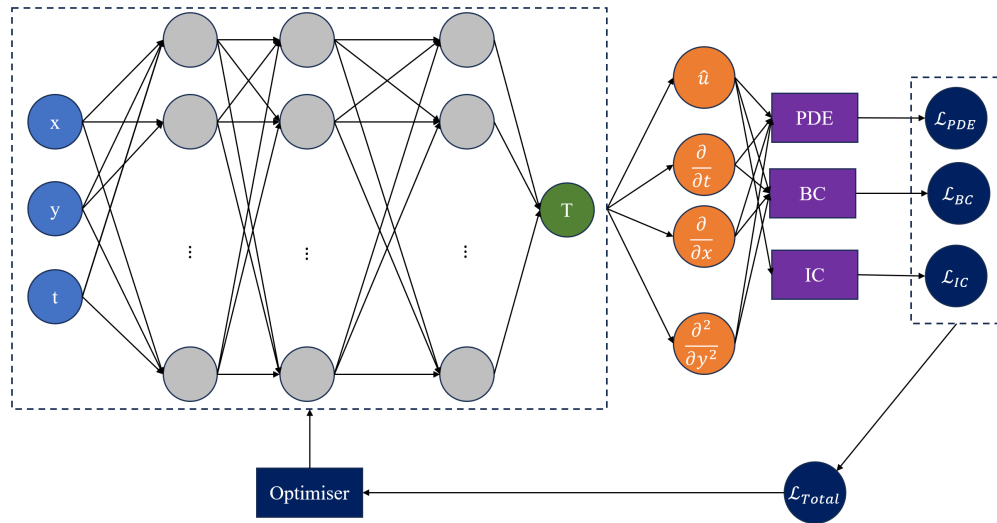


Fig. 2 Architecture of the NN

XPINN, sharing the same NN architecture as PINN, is characterised by the distributed training of multiple PINN sub-networks for pre-defined sub-domains. Figure 3 illustrates the single domain for PINN and the two-sub-domain configuration for XPINN.

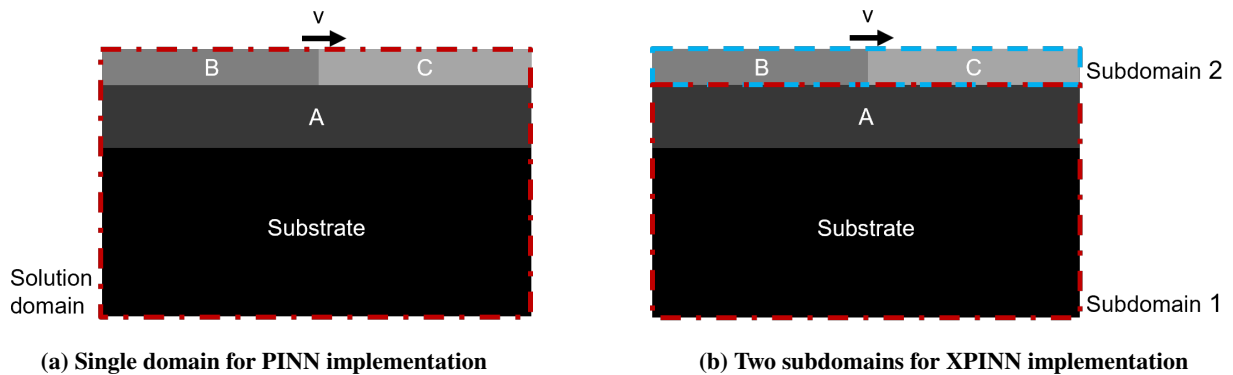


Fig. 3 Illustration of the single domain for PINN and the two subdomains in the two-domain configuration for XPINN

A. Governing equation

Both PINN and XPINN approaches solve the same transient heat conduction to obtain the temperature history with the definition of the heat source model, boundary conditions (BCs), and initial condition (IC) being the key elements. The governing equation(s) for each are defined herein.

The energy equilibrium in an infinitesimal control element over an infinitesimal short period can be expressed as:

$$\frac{\partial(\rho C_p T)}{\partial t} + \nabla \cdot \mathbf{q} = \dot{e}_{gen} \quad (1)$$

where ρ is the density of the material, C_p is the specific heat capacity, T is temperature, \mathbf{q} is heat flux out of the control element, and \dot{e}_{gen} is the body heat source emulating the laser.

Applying Fourier's Law where k is the thermal conductivity and considering density, specific heat capacity, and heat conductivity as temperature-dependent properties,

$$\mathbf{q} = -k(T)\nabla T$$

Equation 1 in the 2D case can be expanded to:

$$T \frac{\partial T}{\partial t} \frac{\partial \rho C_p}{\partial T} + \rho(T) C_p(T) \frac{\partial T}{\partial t} - \frac{\partial k}{\partial T} \left(\left(\frac{\partial T}{\partial x} \right)^2 + \left(\frac{\partial T}{\partial y} \right)^2 \right) - k(T) \left(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} \right) = \dot{e}_{gen} \quad (2)$$

It results in a non-linear problem that is computationally intensive to solve, and a common simplification by assuming a temperature-constant material property is often adopted in the literature which reduces Equation 2 to:

$$\rho C_p \frac{\partial T}{\partial t} - k \left(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} \right) = \dot{e}_{gen} \quad (3)$$

The body heat source (or the 2D equivalent body heat source) is adopted to represent the laser as the laser penetrates into the top layer during the DED process. Hence, it is incorporated into Equation 2 as the heat generation term. The Goldak model [24], commonly used in AM thermal simulations [4], is implemented in the simplified form, and the 2D equivalent can be expressed as:

$$\dot{e}_{gen} = \frac{6\eta\bar{P}}{\pi ab} e^{-3\left(\frac{(x-x_{laser})^2}{a^2} + \frac{(y-y_{laser})^2}{b^2}\right)} \quad (4)$$

where $\bar{P} = \frac{P}{l_c}$ is the laser power per unit length (term l_c is defined in Equation 10), η is the laser absorptivity, c is the radius of the laser spot, b is the penetration depth which is assumed as twice the radius, and x_{laser}, y_{laser} are the centre of the spot.

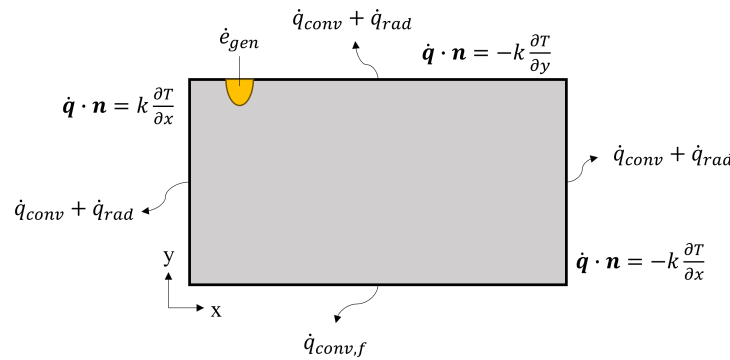


Fig. 4 Assignment of Boundary Conditions

For a unique solution, BCs and IC have to be specified. Neumann BC is employed to describe the energy exchange with the surroundings and the bottom of the substrate as specified in Equation 5. For the bottom, a large convective coefficient h_{force} is applied to emulate the heat transfer from the bottom of the substrate. Figure 4 illustrates the location where different BCs are applied.

$$\mathbf{q}(\mathbf{x}, t) \cdot \mathbf{n} = \sum \dot{q}_{ext}(\mathbf{x}, t), \quad \mathbf{x} \in \Gamma \quad (5)$$

where \mathbf{n} is the normal vector at the boundary and

$$\sum \dot{q}_{ext}(\mathbf{x}, t) = \begin{cases} \dot{q}_{conv} + \dot{q}_{rad} = h(T - T_{amb}) + \sigma \varepsilon (T^4 - T_{amb}^4) & \mathbf{x} \in \Gamma_{left} \cup \Gamma_{right} \cup \Gamma_{top} \\ \dot{q}_{conv, f} = h_{forced}(T - T_{amb}) & \mathbf{x} \in \Gamma_{bottom} \end{cases}$$

The initial condition is treated as Dirichlet:

$$T(\mathbf{x}, 0) = T_{ref}, \quad \mathbf{x} \in \Omega \quad (6)$$

where T_{ref} is defined in a two-part manner due to the layer-by-layer nature of DED. More details on the complications from it are presented in the later section.

B. PINN setup

1. Model Architecture

Fully connected NN, one of the most common types of NN architecture is used for the PINN in this work. The solution from PINN can be generalised as $\hat{T}(\mathbf{x}, t)$ which approximates the target function $T(\mathbf{x}, t)$.

The governing equations for the PDE, BC, and IC discussed in the previous section can be re-arranged and expressed as the following error terms at a given collocation point, respectively:

$$\mathcal{F}(\hat{T}, \mathbf{x}, t) = \rho C_p \frac{\partial \hat{T}}{\partial t} - k \left(\frac{\partial^2 \hat{T}}{\partial x^2} + \frac{\partial^2 \hat{T}}{\partial y^2} \right) - \frac{6\eta \bar{P}}{\pi ab} e^{-3 \left(\frac{(x - (x_0 + vt))^2}{a^2} + \frac{y^2}{b^2} \right)} \quad (7)$$

$$\mathcal{B}(\hat{T}, \mathbf{x}, t) = \begin{cases} \mathbf{q}(\mathbf{x}, t) \cdot \mathbf{n} - (h(\hat{T} - T_{amb}) + \sigma \varepsilon (\hat{T}^4 + T_{amb}^4)) \\ \mathbf{q}(\mathbf{x}, t) \cdot \mathbf{n} - (h_{force}(\hat{T} - T_{amb})) \end{cases} \quad (8)$$

$$\mathcal{I}(\hat{T}, \mathbf{x}, 0) = \hat{T}_{t=t_0} - T_{ref} \quad (9)$$

The governing equations are non-dimensionalised for the two-fold benefits – 1) it eliminates the additional complexity introduced by units; 2) it improves the generalisation capability of the trained model [25]. The non-dimensionalisations are done on the space, time, and temperature scale by introducing the characteristic length l_c , time t_c , and temperature T_c , respectively:

$$\begin{cases} x & = l_c x^* \\ y & = l_c y^* \\ t & = t_c t^* \\ T & = T_c T^* \end{cases} \quad (10)$$

Hence, Equations (7) to (9) can be re-written as:

$$\tilde{\mathcal{F}}(\hat{T}^*, \mathbf{x}^*, t^*) = \frac{\partial \hat{T}^*}{\partial t^*} - \frac{k}{\rho C_p} \frac{t_c}{l_c^2} \left(\frac{\partial^2 \hat{T}^*}{\partial x^{*2}} + \frac{\partial^2 \hat{T}^*}{\partial y^{*2}} \right) - \frac{1}{\rho C_p} \frac{t_c}{T_c} \frac{6\eta\bar{P}}{\pi ab} e^{-3 \left(\frac{(x^* - (x_0^* + v^* t^*))^2 t_c^2}{a^2} + \frac{y^{*2} t_c^2}{b^2} \right)} \quad (11)$$

$$\tilde{\mathcal{B}}(\hat{T}^*, \mathbf{x}^*, t^*) = \begin{cases} \pm \frac{\partial \hat{T}^*}{\partial n^*} - \frac{h}{k} l_c (\hat{T}^* - T_{amb}^*) - \frac{\sigma \varepsilon}{k} T_c^3 l_c (\hat{T}^{*4} + T_{amb}^{*4}) \\ \frac{\partial \hat{T}^*}{\partial y^*} - \frac{h_{force}}{k} l_c (\hat{T}^* - T_{amb}^*) \end{cases} \quad (12)$$

$$\tilde{\mathcal{I}}(\hat{T}^*, \mathbf{x}^*, 0) = \hat{T}_{t=t_0}^* - T_{ref}^* \quad (13)$$

Normalisation of -1 to 1 is applied to the inputs and the output is normalised to a range between 0 and 1 during the training process. It contains the numerical values of the losses to the same order of magnitude, preventing saturating the training due to the large numerical values. Hence, the following characteristic length, time, and temperature are adopted:

$$\begin{cases} l_c = \phi_{laser} \\ t_c = \frac{\phi_{laser}}{v_{laser}} \\ T_c = 3000 \end{cases} \quad (14)$$

The individual loss terms for the training of the PINN can subsequently be written as:

$$\mathcal{L}_{PDE} = \frac{1}{N_{PDE}} \sum_{i=1}^{N_{PDE}} |\tilde{\mathcal{F}}(\hat{T}(\mathbf{x}_{PDE}^i, t_{PDE}^i), \mathbf{x}_{PDE}^i, t_{PDE}^i)|^2 \quad (15)$$

$$\mathcal{L}_{BC} = \frac{1}{N_{BC}} \sum_{i=1}^{N_{BC}} |\tilde{\mathcal{B}}(\hat{T}(\mathbf{x}_{BC}^i, t_{BC}^i), \mathbf{x}_{BC}^i, t_{BC}^i)|^2 \quad (16)$$

$$\mathcal{L}_{IC} = \frac{1}{N_{IC}} \sum_{i=1}^{N_{IC}} |\tilde{\mathcal{I}}(\hat{T}(\mathbf{x}_{IC}^i, 0), \mathbf{x}_{IC}^i, 0, w_{ic}^i)|^2 \quad (17)$$

where N_{PDE} , N_{BC} , and N_{IC} are the number of collocation points sampled to compute the PDE, BC, and IC losses. The w_{ic}^i term in Equation 17 is the pointwise weight for the initial condition loss.

The total loss used for the training of the PINN is obtained by assembling the individual loss terms through a weighted sum:

$$\mathcal{L} = \frac{w_{PDE} \mathcal{L}_{PDE} + w_{BC} \mathcal{L}_{BC} + w_{IC} \mathcal{L}_{IC}}{w_{PDE} + w_{BC} + w_{IC}} \quad (18)$$

The loss expressions are embedded into that NN architecture as illustrated in Figure 2.

2. Spatio-temporal Sampling Points

Collocation points, akin to those in numerical methods, are spatiotemporal sampling points used during the PINN/XPINN training for computing the relevant loss terms to evaluate the level of approximation. When PINN/XPINN is applied as a forward solver without any auxiliary data, the collocation points play a critical role in dictating the convergence and quality of the results. The sensitivity of the training outcome with regard to the distribution and density of collocation points has been well documented in the literature [18, 26, 27]. The weighted total loss in Equation 18 gives a degree of control of the overall relative contributions of the losses. However, it should be noted that the individual loss term is the mean value, implying a need for careful design of the collocation points' density distribution so that the loss values are not biased towards the trivial areas which could occupy the majority of the domain.

In this work, a grid distribution of the collocation points is used for simplicity and ease of definition of the boundaries. Similarly, the temporal domain is uniformly discretised such that the laser spot moves by $\alpha\phi_{laser}$ over each time step. α is chosen to be 0.15 for sufficient overlapping of the sampled collocation points in adjacent timesteps. It should, however, be acknowledged that randomised allocation of the collocation points in the given spatial and temporal domain usually results in better generality [17, 26].

The number of collocation point directly influence the computational time and memory usage during training. Hence, the collocation points are assigned with varying density – global assignment and local assignment. Specifically, for the PDE collocation points, the global region is defined as the overall spatial domain and the local domain is defined as the $(4\phi_{laser} \times 1.5\phi_{laser})$ region centred around and below the centre of the laser spot. To smoothen the transition, a secondary region is defined as that bounded by half the total height from the top and the whole width of the global domain. For BC collocation points, the top side is split into two parts due to the layer-by-layer nature of the DED process - one segment for the top of Zone B and another for the bottom of Zone C in Figure 1. The latter replaces the former as the print progresses. A higher density is applied to the top side compared to the other sides due to the split in the region.

Contrary to other applications of PINN/XPINN as a forward solver commonly seen in the literature, IC collocation points also play an important role in this multi-layer framework since satisfying the IC ensures continuity between the previous and the current layers. In the PINN implementation, the IC collocation points are put into two sets – one for the printed part (Zone A & Substrate) and one for the new layer (Zone B & C). The former is assigned values from the output of the trained model of the completed layer and the latter is assigned the value of ambient temperature since no material is added yet. To ensure enough sampling, the printed zone has a total collocation number of 64 times of that the global PDE domain and the same density is applied to the new layer zone. Such IC allocation is non-conventional and introduces additional challenges to the training. However, it is necessary given the additive nature of the DED process. In the XPINN implementation, the two sets identified in PINN implementation fall nicely into the two subdomains, requiring no further actions.

Figure 5 illustrates the distribution of the generated collocation points for the first layer in printing over the print duration.

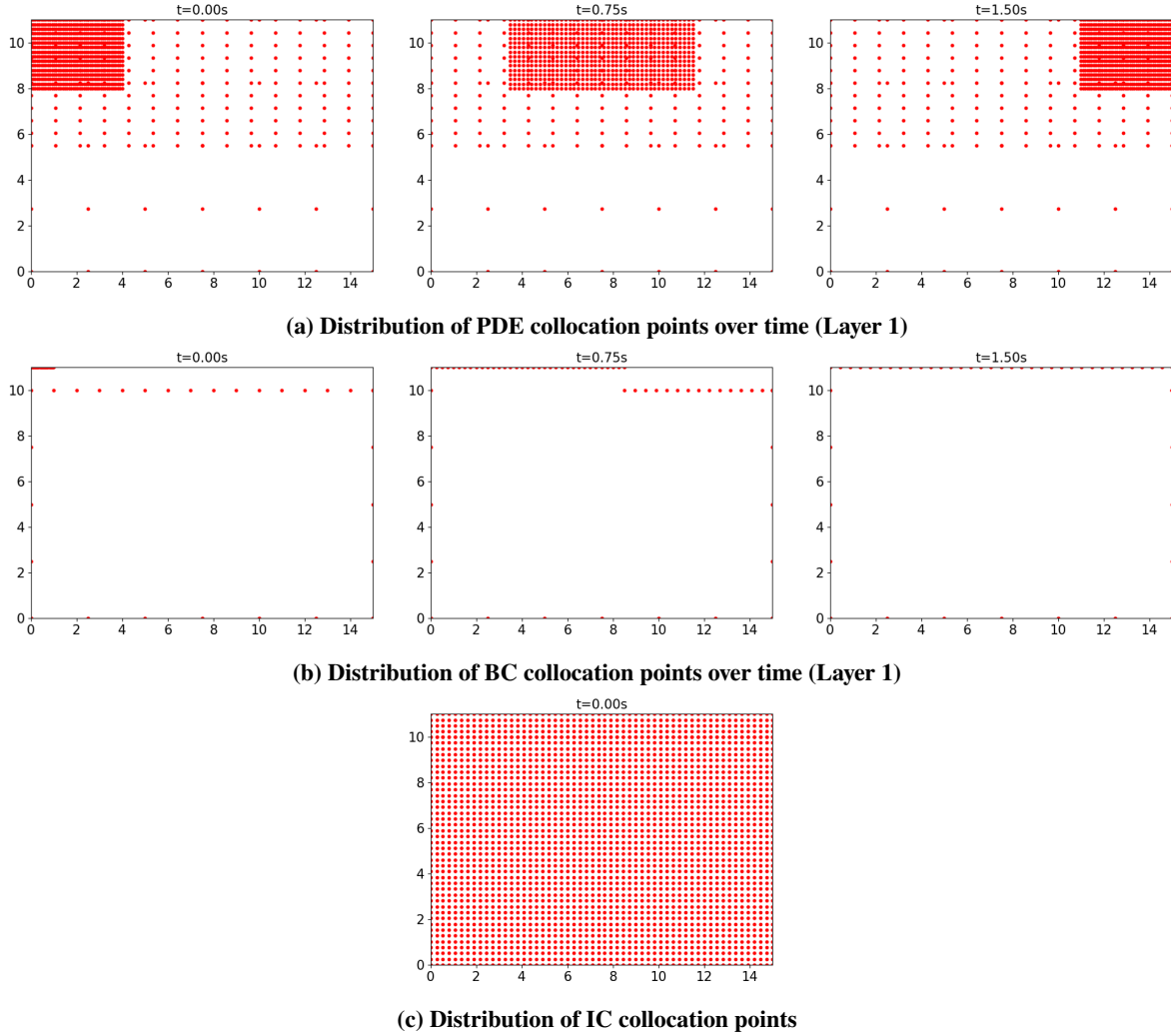


Fig. 5 Distribution of the collocation points for the first layer (Layer 1). The x-axis and y-axis correspond to the length and height of the part domain in a Cartesian coordinate system. The bottom left corner of the part domain is defined as the global origin, indicated by (0,0). The convention is used for all subsequent figures where the part domain is presented.

C. XPINN setup

The setup for XPINN is similar to PINN except for having multiple PINNs for the pre-defined sub-domains, respectively. The sub-domains and the sub-PINNs are integrated through interface conditions which are detailed in [II.C.1](#).

1. Model Architecture

Within each sub-domain, the sub-PINN is defined similarly as those in [Section II.B.1](#) with the same non-dimensionalisation and normalisation applied. Hence, Equation 15-17 can be modified as the following to reflect the definition of subdomains (as indicated by superscript $\square^{(q)}$):

$$\mathcal{L}_{PDE}^{(q)} = \frac{1}{N_{PDE}^{(q)}} \sum_{i=1}^{N_{PDE}^{(q)}} \left| \tilde{\mathcal{F}}^{(q)} \left(\hat{T}^{(q)}(\mathbf{x}_{PDE}^{i,(q)}, t_{PDE}^{i,(q)}), \mathbf{x}_{PDE}^{i,(q)}, t_{PDE}^{i,(q)} \right) \right|^2 \quad (19)$$

$$\mathcal{L}_{BC}^{(q)} = \frac{1}{N_{BC}^{(q)}} \sum_{i=1}^{N_{BC}^{(q)}} \left| \tilde{\mathcal{B}}^{(q)} \left(\hat{T}^{(q)}(\mathbf{x}_{BC}^{i,(q)}, t_{BC}^{i,(q)}), \mathbf{x}_{BC}^{i,(q)}, t_{BC}^{i,(q)} \right) \right|^2 \quad (20)$$

$$\mathcal{L}_{IC}^{(q)} = \frac{1}{N_{IC}^{(q)}} \sum_{i=1}^{N_{IC}^{(q)}} \left| \tilde{\mathcal{I}}^{(q)} \left(\hat{T}^{(q)}(\mathbf{x}_{IC}^{i,(q)}, 0), \mathbf{x}_{IC}^{i,(q)}, 0 \right) \right|^2 \quad (21)$$

Additional interfacial losses are defined as the following (super-script $\square^{(q)}$ indicates the adjacent domain next to subdomain q , and assuming the two subdomains share interface points i.e. $\mathbf{x}^{i,(q)} \equiv \mathbf{x}^{i,(p)}$):

$$\mathcal{L}_{avg}^{(q)} = \frac{1}{N_{int}^{(q)}} \sum_{i=1}^{N_{int}^{(q)}} \left(\hat{T}_i^{(q)} - \frac{\hat{T}_i^{(q)} + \hat{T}_i^{(p)}}{2} \right)^2 \quad (22)$$

$$\mathcal{L}_{PDE_int}^{(q)} = \frac{1}{N_{int}^{(q)}} \sum_{i=1}^{N_{int}^{(q)}} \left| \tilde{\mathcal{F}}^{(q)} \left(\hat{T}^{(q)}(\mathbf{x}^{i,(q)}, t^{i,(q)}), \mathbf{x}^{i,(q)}, t^{i,(q)} \right) - \tilde{\mathcal{F}}^{(p)} \left(\hat{T}^{(p)}(\mathbf{x}^{i,(p)}, t^{i,(p)}), \mathbf{x}^{i,(p)}, t^{i,(p)} \right) \right|^2 \quad (23)$$

$$\mathcal{L}_{dev}^{(q)} = \frac{1}{N_{int}^{(q)}} \sum_{i=1}^{N_{int}^{(q)}} \left(\frac{1}{3} \left(\frac{\partial \hat{T}_i^{(p)}}{\partial x^{i,(q)}} + \frac{\partial \hat{T}_i^{(p)}}{\partial y^{i,(q)}} + \frac{\partial \hat{T}_i^{(p)}}{\partial t^{i,(q)}} \right) \right)^2 \quad (24)$$

For each sub-PINN, the total loss is defined as:

$$\mathcal{L}^{(q)} = \frac{w_{PDE} \mathcal{L}_{PDE}^{(q)} + w_{BC} \mathcal{L}_{BC}^{(q)} + w_{IC} \mathcal{L}_{IC}^{(q)} + w_{avg} \mathcal{L}_{avg}^{(q)} + w_{PDE_int} \mathcal{L}_{PDE_int}^{(q)} + w_{dev} \mathcal{L}_{dev}^{(q)}}{w_{PDE} + w_{BC} + w_{IC} + w_{avg} + w_{PDE_int} + w_{dev}} \quad (25)$$

The total loss for each subdomain is used for the training of each sub-PINN, respectively. The interfacial losses ensure the continuity in solution across adjacent subdomains. Since thermal gradient and cooling rate are also information of interest in this application, continuity in the PDE solution and first-order derivatives are included in training.

2. Spatio-temporal Sampling Points

The spatio-temporal sampling points are defined in the same manner as that in the PINN implementation before being split into the respective sub-domains based on the height of the collocation points. For illustration purposes, in the simplest, 2-subdomain configuration, (Substrate+Zone A) is defined as subdomain 1, and (Zone B+C) is defined as subdomain 2. Any collocation points with a height below the printed layer height are allocated to subdomain 1 and those above are allocated to subdomain 2. It should be highlighted that more subdomains are defined for more complex cases as entailed in the case studies.

Figure 6 illustrates the distribution of the generated collocation points for the first layer over the scan in the two-subdomain configuration.

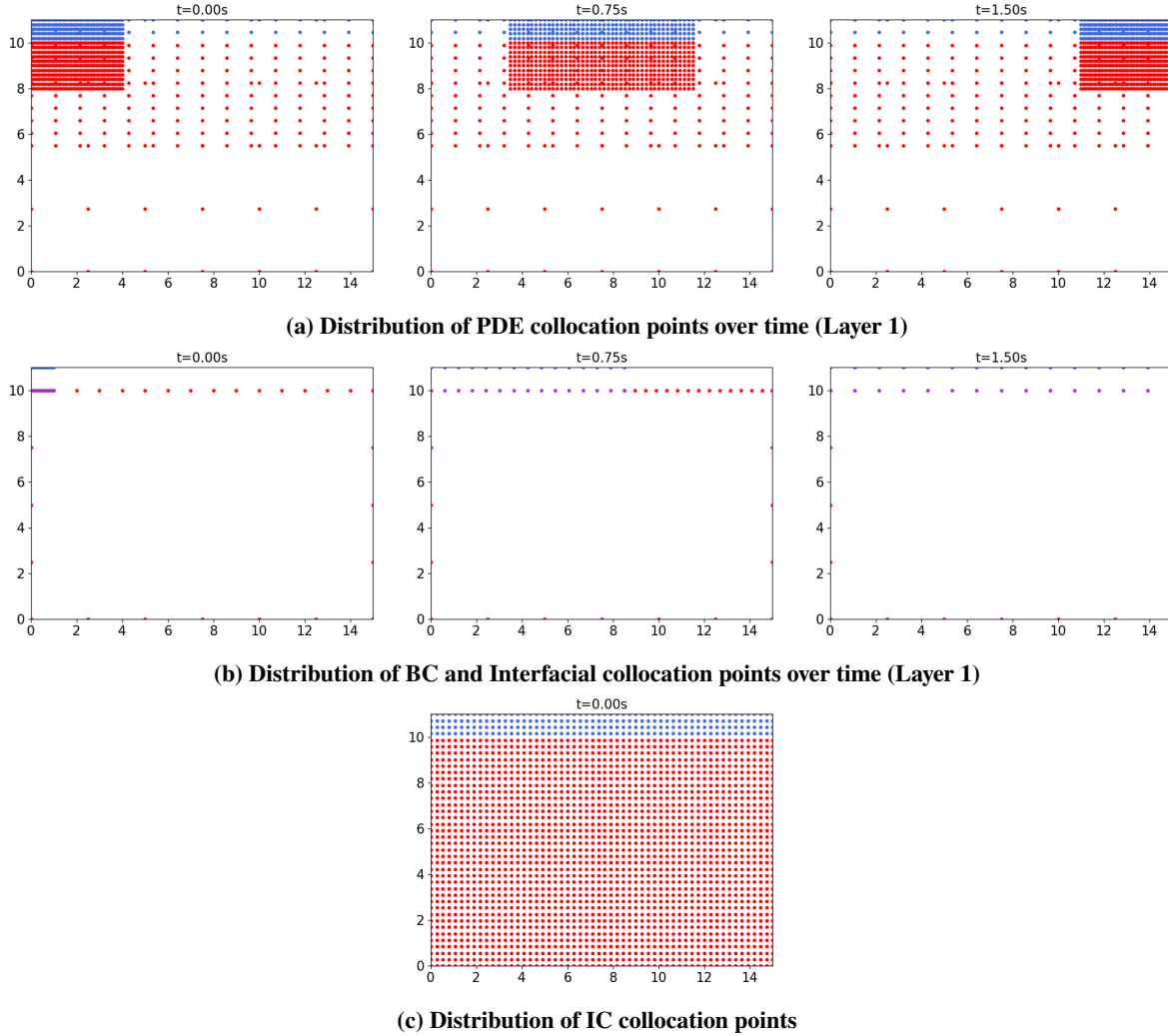


Fig. 6 Distribution of the collocation points for the first layer (Layer 1) in the two-domain XPINN configuration. Red colour indicates subdomain 1, blue colour indicates subdomain 2, and purple colour indicates interfacial collocation points.

D. Multi-scan Framework

A single scan is defined as a path of travel that can be expressed as one continuous function of time with or without on-off switches. It implies that in the 2D case, each layer consists of one scan, assuming that there is no change of direction of the scan within the layer.

Figure 7 illustrates the workflow of the PINN-based and XPINN-based thermal simulation for the multi-scan problem, which is equivalent to a multi-layer problem in 2D.

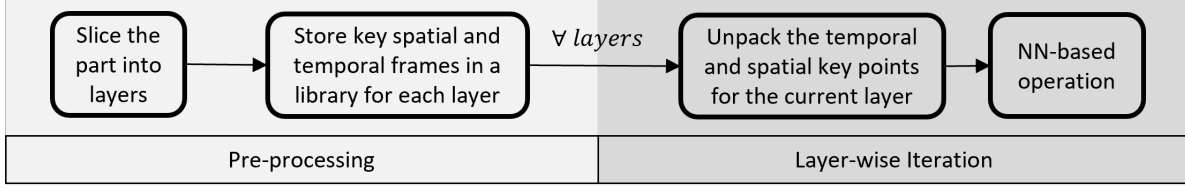


Fig. 7 Simplified flowchart

Pre-processing: The part geometry is sliced into layers. For each layer, the start and end coordinates of the non-part region(s) and subsequently where the laser is switched on and off are determined with regard to the global origin of the whole part domain. The coordinates information is then used to determine the time instances when the laser switches on/off in each scan. The positional information of the part and non-part regions and the temporal information of the laser on/off(s) are stored for the subsequent steps.

Sequentially iterative process: Each sliced layer will be sequentially iterated from the bottom to the top, capturing the printing process.

For each iteration: Each iteration resolves the thermal history associated with one scan which corresponds to one layer in the 2D case. It starts with unpacking the positional and temporal information from the pre-step. The information is used to define terms such as t_{start_i} and t_{end_i} which are used in the functions defined later. A NN of the same architecture for all iterations is initialised with the correct domain size of the current print layer. Collocation points are generated for computing the PDE loss, BC loss, and IC loss, respectively based on the strategies discussed earlier. Depending on the location of the current print layer being simulated, the appropriate initial condition target values are assigned and passed to the loss function. Before starting the training process of the current iteration, a check of the existence of a trained model of a similar scan path in previous layers will be conducted. If it exists, the training will resume after mounting the trained model of the most recent similar scan path. Otherwise, the training will start from the standard initialisation. The training in each iteration will stop when any stopping criterion (either the maximum epoch number or the threshold loss value) is met.

To accurately model the DED process, several strategies are incorporated to reduce the amount of discontinuity and ease the process of training:

1. Laser on-off switch

The on-off state of the laser is dictated by the temporal marker(s) during each scan which are determined based on the slice of geometry of the part for each layer. Hence, the original heat source equation (Equation 4) is modified to Equation 26 and the laser switch function $laser_switch(t)$ is expressed as Equation 27.

$$q_{laser} = laser_switch(t) \frac{6\eta\bar{P}}{\pi ab} \exp\left(-3\left(\frac{(x - x_{laser})^2}{a^2} + \frac{(y - y_{laser})^2}{b^2}\right)\right) \quad (26)$$

$$laser_switch(t) = \sum_i^N \left(\frac{1}{1 + e^{-\sigma(t-t_{start_i})}} - \frac{1}{1 + e^{-\sigma(t-t_{stop_i})}} \right) \quad (27)$$

where N is the total instances of switching on the laser (i.e. $N \geq 1$). Parameter σ defines how quickly the switching between On and Off occurs, and a value of 200 is used in this work for a balance between rise time and smoothness.

2. Solid-void transition

The diffusive property of air needs to be explicitly assigned to Zone C for the correct amount of heat transfer to be captured and to avoid the region behaving as an infinite heat sink. A smooth transition between the solid (Zone A & B) and void (Zone C) is achieved through the use of sigmoid function and can be expressed as:

$$\begin{aligned} \tilde{\mathcal{F}} = & \tilde{\mathcal{F}}^{solid} \left(\left[\left(1 + e^{\alpha_z (x^* - x_{boundary}^*)} \right) \left(1 + e^{-\alpha_z (y^* - y_{boundary}^*)} \right) \right]^{-1} + \left[1 + e^{\alpha_z (y^* - y_{boundary}^*)} \right]^{-1} \right) \\ & + \tilde{\mathcal{F}}^{air} \left[\left(1 + e^{-\alpha_z (x^* - x_{boundary}^*)} \right) \left(1 + e^{-\alpha_z (y^* - y_{boundary}^*)} \right) \right]^{-1} \end{aligned} \quad (28)$$

where α_z controls how steep the transition is and a value of 100 is chosen.

It allows a regular grid of collocation points to be generated for the PDE loss for the whole domain while achieving a smooth transition in distinguishing the solid and air regions.

E. Training with causality

Incorporating causality in the training is proposed in [28] where the authors show that standard training of PINN is ‘implicitly biased towards first approximating PDE solutions at later times’, necessitating a training strategy that focuses on satisfying the solution in the correct temporal sequence (i.e. causality). It is identified that causality has a critical role in the multilayer problem since the correct solution of each subsequent layer depends on correctly capturing the initial condition which itself is challenging as detailed earlier.

The key component to incorporate causality is by introducing a ‘causal weight array’ which is defined as:

$$w_c^i = e^{-\epsilon (\sum_{k=1}^{i-1} \mathcal{L}_t(t_k, \mathbf{x}) + w_{IC} \mathcal{L}_{IC})} \quad \text{for } i = 2, 3, \dots, N_t \quad (29)$$

where ϵ is a hyperparameter controlling the stiffness of the causality, N_t is the number of timesteps where the collocation points are generated and L_t is defined as a N_t -element array that is expressed as:

$$\mathcal{L}_t^i = w_{PDE} \mathcal{L}_{PDE}(t^i, \mathbf{x}_{PDE}^i) + w_{BC} \mathcal{L}_{BC}(t^i, \mathbf{x}_{BC}^i) \quad \text{for } i = 2, 3, \dots, N_t \quad (30)$$

It implies that the expression for the cost of training is modified to:

$$\mathcal{L} = \frac{1}{N_t} \sum_{i=0}^{N_t} \left(\frac{w_c^i}{w_{PDE} + w_{BC} + w_{IC}} \mathcal{L}_t^i + \frac{w_{IC}}{w_{PDE} + w_{BC} + w_{IC}} \mathcal{L}_{IC} \right) \quad (31)$$

F. Benchmark from ANSYS

The numerical benchmark is generated by running a detailed thermal simulation in ANSYS. The same implementation has been adopted for calibration in MIS-related works by Prof. Albert To’s group [8–12]. The same boundary conditions and Goldak heat source model as entailed in Section II.A are applied in the ANSYS Mechanical simulation which utilises the element activation method. The time history of each element is extracted during the post-processing step and the spatiotemporal coordinates are used to generate the prediction from the training PINN for comparison. Both root-mean-square error (RMSE) and maximum absolute error are extracted to measure the accuracy of the output.

For benchmarking, mesh resolutions of 0.15mm in length, 0.05mm in height, 0.63mm in substrate height, and a time resolution of 0.0075s is applied.

G. Case Study

The comparison between the PINN-base and XPINN-based implementation focuses on two main comparison metrics: 1) prediction accuracy as compared to the ANSYS benchmark result; 2) computational effort required for the training and solution.

The printed part and the substrate form a simple rectangular domain for the simulation. A three-layer scan of a rectangular design without interpass time is investigated and illustrated in Figure 8. Spatial decomposition is applied in the XPINN implementation. The three layers are printed in alternating laser directions.

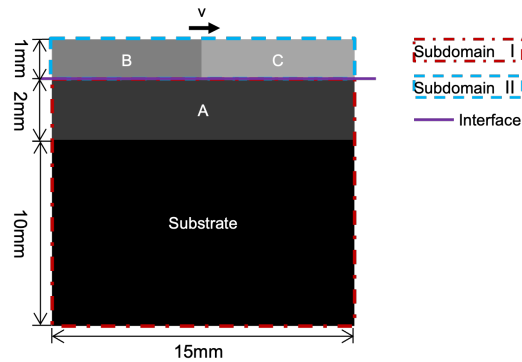


Fig. 8 Illustration of Case 1 and the spatial decomposition. Image showing the last layer and not drawn to scale.

III. Results & Discussion

The comparison between the outputs from the PINN-based and the XPINN-based frameworks is visualised in Figure 9 and Figure 10.

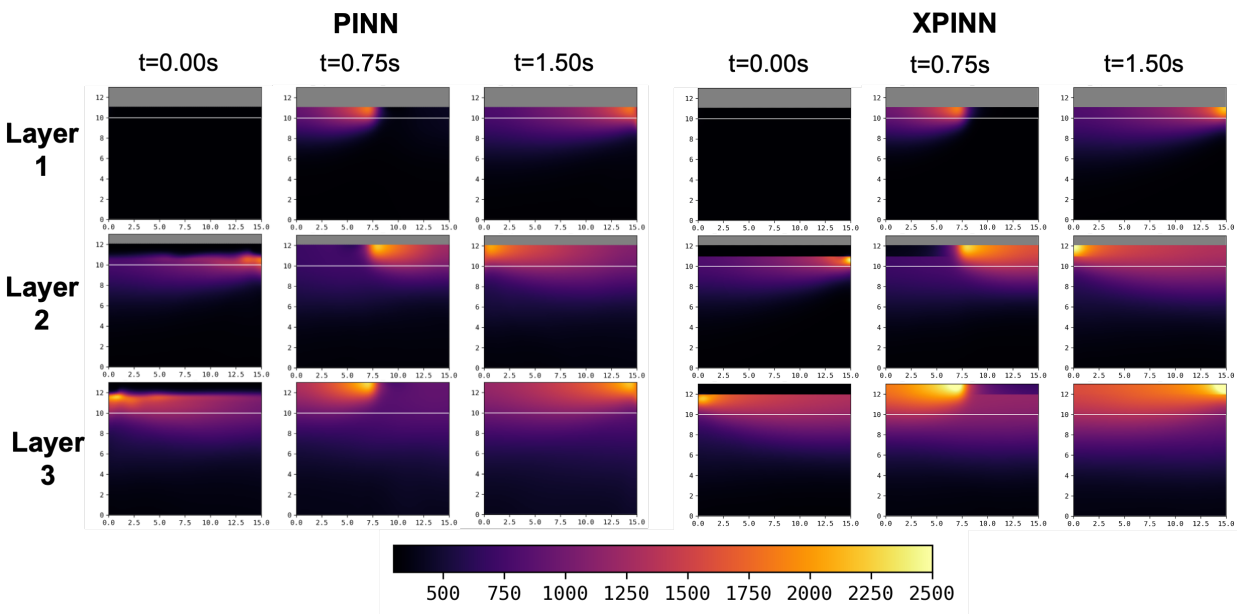


Fig. 9 Temperature output comparison between PINN and XPINN for Case 1.

Table 1 summarises the quantitative comparison between the PINN- and XPINN-based frameworks of their

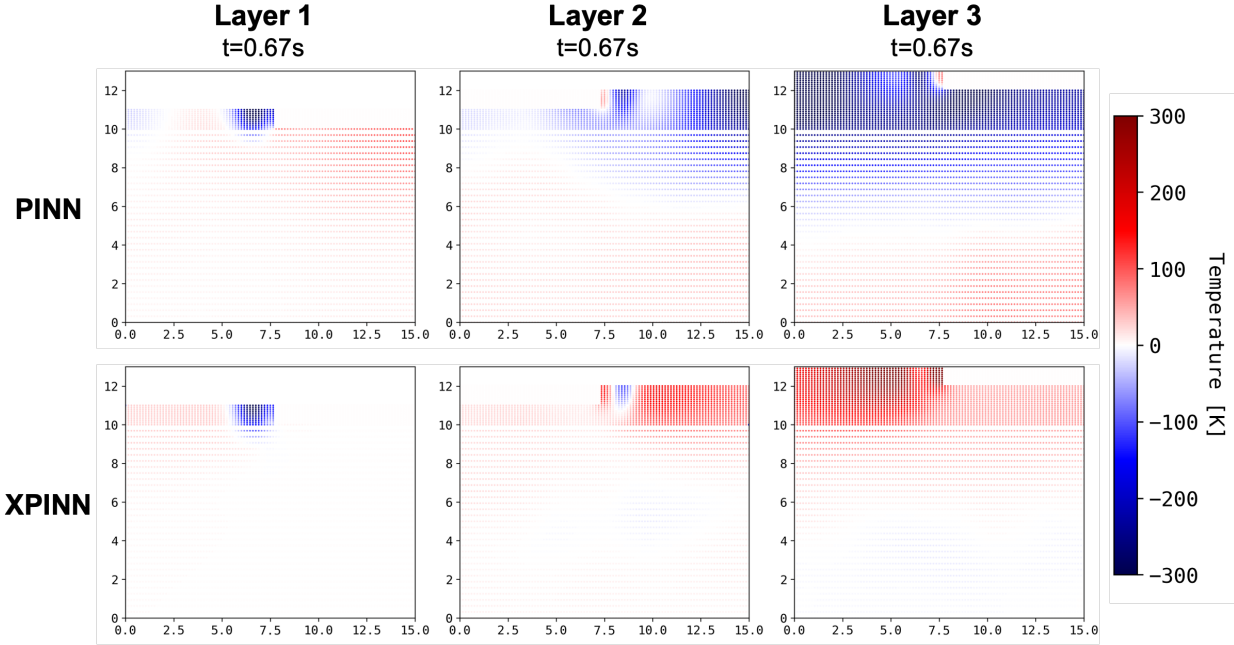


Fig. 10 Visualisation of benchmark error against ANSYS solution for both PINN and XPINN. Showing the snapshot at $t=0.67s$ of each layer. Benchmark error is defined as $(T_{XPINN \text{ or } PINN} - T_{ANSYS})$ for all plots in this paper unless otherwise stated.

performance.

Table 1 Quantitative performance comparison between PINN and XPINN-based frameworks

		RMSE	Max. Error	Training
		[K]	[K]	Time [s]
Layer 1	PINN	22.36	1160.25	4349
	XPINN	10.98	553.32	14228
Layer 2	PINN	51.25	1163.63	4285
	XPINN	28.76	503.23	14403
Layer 3	PINN	126.89	1337.41	4207
	XPINN	65.10	642.63	14383

XPINN achieves significant accuracy improvement compared to the PINN-implementation. It can largely be attributed to the spatial decomposition – the subnet for the top layer is dedicated to capturing the localised effect of the heat source while the subnet for the printed layers and the substrate is dedicated to capturing the heat dissipation in the whole domain. The separation of the domains allows the overall XPINN to be able to capture phenomena on multiple scales which is challenging for a single neural network used in the PINN implementation. It should also be acknowledged that without parallelising the two subnets, the computational effort approximately doubles compared to that for the PINN framework as a result of two backpropagations and the subsequent optimisations being run in the XPINN implementation. The increase in computational time may be resolved by running the training of the subnets on multiple GPUs concurrently, a capability that has been demonstrated [29]. However, it is beyond the scope of this work. Nevertheless, the observation that multiple subnets incur extra computational cost raises the need to be prudent when decomposing domains, as discussed further in the later section.

IV. Concluding Remarks and Next Steps

In this work, we present an XPINN-based framework that furthers the initial PINN-based implementation [20] in predicting temperature history during the DED process. Leveraging domain decomposition, the XPINN-based framework is able to achieve improved prediction accuracy, around 50% reduction in RMSE and absolute error. Having demonstrated the effectiveness of domain decomposition by XPINN in accounting for discontinuities in the solution domain, it is expected that XPINN will be an effective tool for accounting for more complex geometries and print configurations. The case studies have been undertaken and full details have recently been published in [30]. For simplicity in explanation and discussion, the 2D implementation is presented herein. Extension of the frameworks to 3D is easily achievable following the same procedure presented in [20] in detail. Two main directions for future work are identified to further improve the proposed XPINN-based framework: 1) computational time reduction through multi-GPU parallelisation where the training of the subnets is distributed and parallelised. 2) performance improvement by hyper-parameter tuning. With the increased number of NNs in XPINN, hyperparameter tuning for XPINN is a non-trivial task and could potentially benefit from techniques such as meta-learning and Bayesian Optimisation.

References

- [1] Vaidya, S., Ambad, P., and Bhosle, S., “Industry 4.0 – A Glimpse,” *Procedia Manufacturing*, Vol. 20, 2018, pp. 233–238. <https://doi.org/10.1016/j.promfg.2018.02.034>, URL <https://www.sciencedirect.com/science/article/pii/S2351978918300672>.
- [2] Frazier, W. E., “Metal Additive Manufacturing: A Review,” *Journal of Materials Engineering and Performance*, Vol. 23, No. 6, 2014, pp. 1917–1928. <https://doi.org/10.1007/s11665-014-0958-z>, URL <https://doi.org/10.1007/s11665-014-0958-z>.
- [3] Ramoni, M., Shanmugam, R., Thangapandian, N., and Vishnuvarthanan, M., “Challenges in Additive Manufacturing for Metals and Alloys,” *Innovations in Additive Manufacturing*, edited by M. A. Khan and J. T. W. Jappes, Springer Tracts in Additive Manufacturing, Springer International Publishing, Cham, 2022, pp. 57–72. https://doi.org/10.1007/978-3-030-89401-6_3, URL https://doi.org/10.1007/978-3-030-89401-6_3.
- [4] Bayat, M., Dong, W., Thorborg, J., To, A. C., and Hattel, J. H., “A review of multi-scale and multi-physics simulations of metal additive manufacturing processes with focus on modeling strategies,” *Additive Manufacturing*, Vol. 47, 2021, p. 102278. <https://doi.org/10.1016/j.addma.2021.102278>, URL <https://linkinghub.elsevier.com/retrieve/pii/S2214860421004383>.
- [5] Denlinger, E. R., Gouge, M., Irwin, J., and Michaleris, P., “Thermomechanical model development and in situ experimental validation of the Laser Powder-Bed Fusion process,” *Additive Manufacturing*, Vol. 16, 2017, pp. 73–80. <https://doi.org/10.1016/j.addma.2017.05.001>, URL <https://www.sciencedirect.com/science/article/pii/S2214860417300441>.
- [6] Parry, L., Ashcroft, I. A., and Wildman, R. D., “Understanding the effect of laser scan strategy on residual stress in selective laser melting through thermo-mechanical simulation,” *Additive Manufacturing*, Vol. 12, 2016, pp. 1–15. <https://doi.org/10.1016/j.addma.2016.05.014>, URL <https://www.sciencedirect.com/science/article/pii/S2214860416300987>.
- [7] Cheng, B., Shrestha, S., and Chou, K., “Stress and deformation evaluations of scanning strategy effect in selective laser melting,” *Additive Manufacturing*, Vol. 12, 2016, pp. 240–251. <https://doi.org/10.1016/j.addma.2016.05.007>, URL <https://www.sciencedirect.com/science/article/pii/S2214860416300914>.
- [8] Liang, X., Cheng, L., Chen, Q., Yang, Q., and To, A. C., “A modified method for estimating inherent strains from detailed process simulation for fast residual distortion prediction of single-walled structures fabricated by directed energy deposition,” *Additive Manufacturing*, Vol. 23, 2018, pp. 471–486. <https://doi.org/10.1016/j.addma.2018.08.029>, URL <https://linkinghub.elsevier.com/retrieve/pii/S2214860418304858>.
- [9] Chen, Q., Liang, X., Hayduke, D., Liu, J., Cheng, L., Oskin, J., Whitmore, R., and To, A. C., “An inherent strain based multiscale modeling framework for simulating part-scale residual deformation for direct metal laser sintering,” *Additive Manufacturing*, Vol. 28, 2019, pp. 406–418. <https://doi.org/10.1016/j.addma.2019.05.021>, URL <https://linkinghub.elsevier.com/retrieve/pii/S2214860418310583>.
- [10] Dong, W., Liang, X., Chen, Q., Hinnebusch, S., Zhou, Z., and To, A. C., “A new procedure for implementing the modified inherent strain method with improved accuracy in predicting both residual stress and deformation for laser powder bed fusion,” *Additive Manufacturing*, Vol. 47, 2021, p. 102345. <https://doi.org/10.1016/j.addma.2021.102345>, URL <https://linkinghub.elsevier.com/retrieve/pii/S2214860421005030>.
- [11] Liang, X., Dong, W., Chen, Q., and To, A. C., “On incorporating scanning strategy effects into the modified inherent strain modeling framework for laser powder bed fusion,” *Additive Manufacturing*, Vol. 37, 2021, p. 101648. <https://doi.org/10.1016/j.addma.2020.101648>, URL <https://linkinghub.elsevier.com/retrieve/pii/S2214860420310204>.
- [12] Dong, W., Jimenez, X. A., and To, A. C., “Temperature-dependent modified inherent strain method for predicting residual stress and distortion of Ti6Al4V walls manufactured by wire-arc directed energy deposition,” *Additive Manufacturing*, Vol. 62, 2023, p. 103386. <https://doi.org/10.1016/j.addma.2022.103386>, URL <https://linkinghub.elsevier.com/retrieve/pii/S2214860422007758>.
- [13] Demir, K., Zhang, Z., Ben-Artzy, A., Hosemann, P., and Gu, G. X., “Laser scan strategy descriptor for defect prognosis in metal additive manufacturing using neural networks,” *Journal of Manufacturing Processes*, Vol. 67, 2021, pp. 628–634. <https://doi.org/10.1016/j.jmapro.2021.05.011>, URL <https://www.sciencedirect.com/science/article/pii/S1526612521003339>.
- [14] Ren, K., Chew, Y., Zhang, Y. F., Fuh, J. Y. H., and Bi, G. J., “Thermal field prediction for laser scanning paths in laser aided additive manufacturing by physics-based machine learning,” *Computer Methods in Applied Mechanics and Engineering*, Vol. 362, 2020, p. 112734. <https://doi.org/10.1016/j.cma.2019.112734>, URL <https://www.sciencedirect.com/science/article/pii/S0045782519306243>.

- [15] Hong, R., Zhang, L., Lifton, J., Daynes, S., Wei, J., Feih, S., and Lu, W. F., “Artificial neural network-based geometry compensation to improve the printing accuracy of selective laser melting fabricated sub-millimetre overhang trusses,” *Additive Manufacturing*, Vol. 37, 2021, p. 101594. <https://doi.org/10.1016/j.addma.2020.101594>, URL <https://www.sciencedirect.com/science/article/pii/S2214860420309660>.
- [16] Ogoke, F., Lee, W., Kao, N.-Y., Myers, A., Beuth, J., Malen, J., and Barati Farimani, A., “Convolutional neural networks for melt depth prediction and visualization in laser powder bed fusion,” *The International Journal of Advanced Manufacturing Technology*, 2023. <https://doi.org/10.1007/s00170-023-12384-z>, URL <https://doi.org/10.1007/s00170-023-12384-z>.
- [17] Raissi, M., Perdikaris, P., and Karniadakis, G. E., “Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations,” *Journal of Computational Physics*, Vol. 378, 2019, pp. 686–707. <https://doi.org/10.1016/J.JCP.2018.10.045>.
- [18] Liao, S., Xue, T., Jeong, J., Webster, S., Ehmann, K., and Cao, J., “Hybrid thermal modeling of additive manufacturing processes using physics-informed neural networks for temperature prediction and parameter identification,” *Computational Mechanics*, Vol. 72, No. 3, 2023, pp. 499–512. <https://doi.org/10.1007/s00466-022-02257-9>, URL <https://link.springer.com/10.1007/s00466-022-02257-9>.
- [19] Zhu, Q., Liu, Z., and Yan, J., “Machine learning for metal additive manufacturing: predicting temperature and melt pool fluid dynamics using physics-informed neural networks,” *Computational Mechanics*, Vol. 67, No. 2, 2021, pp. 619–635. <https://doi.org/10.1007/s00466-020-01952-9>, URL <https://link.springer.com/10.1007/s00466-020-01952-9>.
- [20] Peng, B., and Panesar, A., “Multi-layer thermal simulation using physics-informed neural network,” *Additive Manufacturing*, Vol. 95, 2024, p. 104498. <https://doi.org/10.1016/j.addma.2024.104498>, URL <https://www.sciencedirect.com/science/article/pii/S221486042400544X>.
- [21] Karniadakis, A. D. J. . G. E., “Extended Physics-Informed Neural Networks (XPINNs): A Generalized Space-Time Domain Decomposition Based Deep Learning Framework for Nonlinear Partial Differential Equations,” *Communications in Computational Physics*, Vol. 28, No. 5, 2020, pp. 2002–2041. <https://doi.org/10.4208/cicp.OA-2020-0164>, URL http://global-sci.org/intro/article_detail/cicp/18403.html.
- [22] Hu, Z., Jagtap, A. D., Karniadakis, G. E., and Kawaguchi, K., “When Do Extended Physics-Informed Neural Networks (XPINNs) Improve Generalization?” *SIAM Journal on Scientific Computing*, Vol. 44, No. 5, 2022, pp. A3158–A3182. <https://doi.org/10.1137/21M1447039>, URL <https://epubs.siam.org/doi/10.1137/21M1447039>, 28 citations (Crossref) [2024-01-02] Read_Status: Read Read_Status_Date: 2024-01-15T16:33:40.775Z rate: 4.
- [23] Baydin, A. G., Pearlmutter, B. A., Radul, A. A., and Siskind, J. M., “Automatic differentiation in machine learning: a survey,” 2015.
- [24] Goldak, J., Chakravarti, A., and Bibby, M., “A new finite element model for welding heat sources,” *Metallurgical Transactions B*, Vol. 15, No. 2, 1984, pp. 299–305. <https://doi.org/10.1007/BF02667333>, URL <https://link.springer.com/10.1007/BF02667333>.
- [25] Wang, S., Sankaran, S., Wang, H., and Perdikaris, P., “An Expert’s Guide to Training Physics-informed Neural Networks,” , Aug. 2023. URL <http://arxiv.org/abs/2308.08468>, arXiv:2308.08468 [physics] keywords = Computer Science - Machine Learning, Mathematics - Numerical Analysis, Physics - Computational Physics, PINN..
- [26] Cuomo, S., Di Cola, V. S., Giampaolo, F., Rozza, G., Raissi, M., and Piccialli, F., “Scientific Machine Learning Through Physics-Informed Neural Networks: Where we are and What’s Next,” *Journal of Scientific Computing*, Vol. 92, No. 3, 2022, p. 88. <https://doi.org/10.1007/s10915-022-01939-z>, URL <https://link.springer.com/10.1007/s10915-022-01939-z>.
- [27] Subramanian, S., Kirby, R. M., Mahoney, M. W., and Gholami, A., “Adaptive Self-supervision Algorithms for Physics-informed Neural Networks,” , Jul. 2022. <https://doi.org/10.48550/arXiv.2207.04084>, URL <http://arxiv.org/abs/2207.04084>, arXiv:2207.04084 [physics].
- [28] Wang, S., Sankaran, S., and Perdikaris, P., “Respecting causality is all you need for training physics-informed neural networks,” , Mar. 2022. URL <http://arxiv.org/abs/2203.07404>, arXiv:2203.07404 [nlin, physics:physics, stat] keywords = Computer Science - Machine Learning, Statistics - Machine Learning, Mathematics - Numerical Analysis, Nonlinear Sciences - Chaotic Dynamics, Physics - Fluid Dynamics, PINN.
- [29] Shukla, K., Jagtap, A. D., and Karniadakis, G. E., “Parallel physics-informed neural networks via domain decomposition,” *Journal of Computational Physics*, Vol. 447, 2021, p. 110683. <https://doi.org/10.1016/j.jcp.2021.110683>, URL <https://www.sciencedirect.com/science/article/pii/S0021999121005787>.
- [30] Peng, B., and Panesar, A., “Multi-layer thermal history prediction framework for directed energy deposition based on extended physics-informed neural networks (XPINN),” *Additive Manufacturing*, Vol. 110, 2025, p. 104953. <https://doi.org/10.1016/j.addma.2025.104953>, URL <https://www.sciencedirect.com/science/article/pii/S2214860425003173>.