

Multi-layer thermal simulation using physics-informed neural network

Bohan Peng^a, Ajit Panesar^{a,*}

^aDepartment of Aeronautics, Imperial Collge London, South Kensington Campus, Exhibition Road, London, SW7 2AZ, United Kingdom

ARTICLE INFO

Keywords:

Physics-informed neural network
DED
Thermal simulation
Scientific Machine Learning

ABSTRACT

This paper presents a physics-informed neural network (PINN)-based solution framework that predicts the thermal history during a multi-layer Directed Energy Deposition (DED) process. The meshless nature and the readily available derivative information of PINN solution opens up new opportunities for modelling the thermally induced distortion in metal Additive Manufacturing (AM). The proposed framework incorporates simple yet effective strategies that enable PINN to overcome the usual shortfall of neural networks (NNs) in dealing with discontinuities. It is a critical step for applying PINN to the multi-layer problem which intrinsically contains discontinuities due to the layer-by-layer nature of DED and other metal AM processes. The accuracy of the proposed framework is validated via a benchmark test against ANSYS simulation. Leveraging the possibility of initialisation with prior knowledge, PINN is also demonstrating potential computational time-savings, especially for larger parts. Furthermore, remarks on strategies to improve ease of training and prediction accuracy by PINN for the particular use case in DED temperature history prediction have been made. The proposed framework sets the foundation for the subsequent exploration of applying scientific machine learning (SciML) techniques to real-life engineering applications.

1. 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 Directed 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 predictions, regardless of being metallurgical or mechanical, are usually a two-step process due to their weakly-coupled nature. The first step is usually a thermal analysis of the process which involves two main schools of methods – conduction-based and CFD-based. Due to the computational cost and recent progress in computing power, the latter has been gaining momentum over recent years while the former is more commonly applied, especially for analyses at the part-scale. The conduction-based thermal model is commonly used for both meso- and part-scale thermomechanical simulation. At the mesoscale, the scan paths are resolved explicitly which captures the effect of the scan pattern on residual stress/strain but it results in high computational cost. In the work by Denlinger et al. [5], they managed to obtain the thermo-elasto-plastic evolution during the L-PBF process. However, it is only up to 38 layers of a cube. Attempts are made by other researchers such as Olleak and Xi [6] and Moran et al. [7] to push the analyses to the part level. However, their analyses only involve the thermal component. The attempts of mesoscale, conduction-based analyses

*Corresponding author

 a.panesar@imperial.ac.uk (A. Panesar)
ORCID(s): 0000-0001-8801-8288 (A. Panesar)

are usually based on finite element analyses (FEA) and make use of techniques such as element coarsening, adaptive re-meshing etc. to speed up the process. By neglecting certain thermal phenomena such as phase transformation, Parry et al. [8] and Cheng et al. [9] 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. The thermomechanical simulations at the part scale usually aim to obtain the overall part distortion. By homogenising the scan patterns over multiple layers, quick computation can be achieved with a reasonable level of accuracy after proper calibration for the material and process. A series of works based on the modified inherent strain (MIS) method have been published [10, 11, 12, 13, 14]. Instead of explicitly solving for the thermal history, MIS-based methods usually have a detailed simulation stage to calibrate the inherent strain during printing and use the information to predict the overall part distortion. The accuracy of the MIS methods has improved with more phenomena captured progressively. In Dong et al.'s work [14], thermal simulation is re-introduced since the heat accumulation during large-scale DED has too large an impact to be neglected and the accuracy improvement outweighs the increased computational cost.

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 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 [15]. Also related to the 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 [16]. Another use of NN is to generate compensation for overhang angle to improve the print accuracy of lattice structures through MAM [17]. 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 [18]. 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) [19] 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. When a forward problem is solved with PINN, while it is often termed as 'prediction', there are two classes of applications that should be distinguished:

1. Class 1: No data is used over the temporal domain for the solution. Hence, the trained model approximates the solution as a solver and predicts the evolution of the output(s). It is categorised as self-supervised learning.
2. Class 2: Labelled data (e.g. temperature values) is added into the PINN framework, usually in the form of a data loss term, over the entirety or majority of the temporal domain, making it semi-supervised learning. Many works in the literature are in this class.

Many attempts of using PINN for temperature history simulation are single-layer. [20] applied PINN to predict the melt pool dimension akin to the method in [21]. The authors then attempted to predict the temperature field for a single scan. [22] applies PINN for the Class 1 forward problem and demonstrates the effectiveness of incorporating residue block in the NN and initialising from a trained model. The method is restricted to the single-layer problem. There are some works attempting the multi-layer DED problem. In [23], training for both single-layer and multi-layer requires labelled data. Hence, it is a Class 2 application of PINN in the forward problem and a substantial portion of the article is to justify the accuracy of the source of data. It neglects the discontinuity in the initial condition between the new and already-printed layers for the multi-layer problem. 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 [24]. The multi-layer problem is briefly presented. With the temperature data over the scan duration utilised, the multi-layer attempt remains Class 2 of the forward problem.

In this work, we propose a multi-layer framework that emulates the DED manufacturing process, allowing part-scale, meshless thermal history prediction based on PINN. The authors envision that the proposed framework lays the foundation for the progression from geometry-only to process-aware design optimisation for DED whose temperature history is intrinsically discontinuous with its layer-by-layer nature – i.e. as the deposition of the new layer is simulated, there is a discontinuity of temperature field between the previously deposited layer and the region where the new layer is to be deposited (region of air with room temperature). It poses significant challenges to the multi-layer simulation using NN-based methods since NN does not handle discontinuity well despite the persisting endeavour among researchers.

One key contribution of this work is the proposed strategies that make PINN contiguous with the discontinuities in simulating the multi-layer DED process while remaining as a Class 1 application of PINN (i.e. without labelled data over the simulated duration). Computational gains supported by initialisation based on a previously trained model, especially for large parts, are also explored. The paper is structured as follows. The multi-layer framework and relevant building blocks are first presented in detail together with the definition of several test cases used to evaluate the effect of the building blocks and investigate the ML-specific strategies. The results corresponding to the test cases are presented and discussed subsequently. It is followed by a benchmark study demonstrating the performance of the proposed method compared to ANSYS. The unique super-resolution capability from PINN's meshless nature and the readily available derivative information for the solution are also illustrated. Finally, concluding remarks are made, accompanied by the outline of future work.

2. Methods

PINN, a type of SciML-based method, is used as a forward solver for the temperature history during the DED process and Figure 1 illustrates the architecture of the NN employed. At the core of PINN, the auto-differentiation [25] 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.

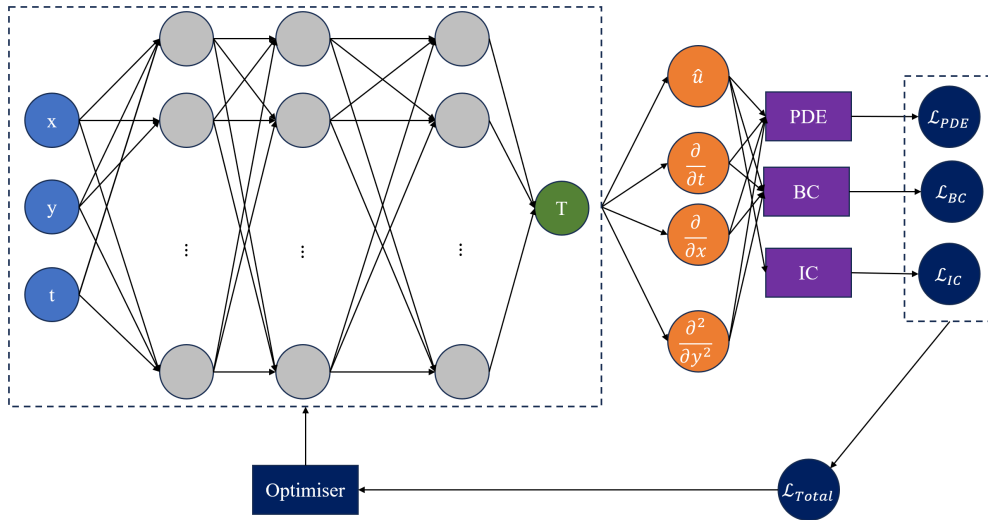


Figure 1: Architecture of the NN

2.1. Governing equation

Transient heat conduction is solved 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. For brevity, the equations and the relevant expansions are elaborated in their 2D form in this section. The 3D implementation is attached in Appendix A for completeness and any interest from the reader.

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

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

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 $\dot{e}_{\text{gen}}$ 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:

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

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:

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

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 [26], commonly used in AM thermal simulations [4], is implemented in the simplified form, and the 2D equivalent can be expressed as:

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

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, a is the radius of the laser spot, b is the penetration depth which is assumed as twice the radius, and $x_{\text{laser}}, y_{\text{laser}}$ are the centre of the spot.

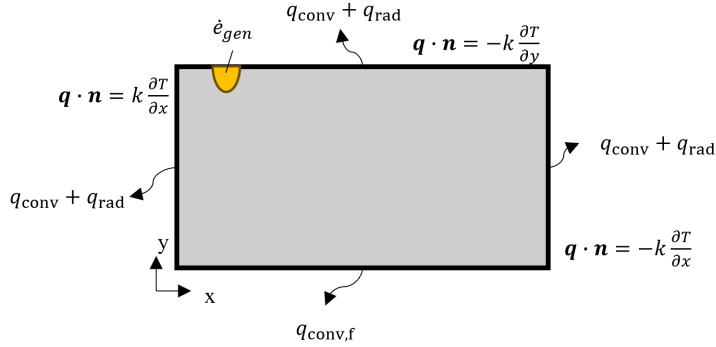


Figure 2: Assignment of Boundary Conditions

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 2 illustrates the location where different BCs are applied.

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

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

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

The initial condition is treated as Dirichlet:

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

where T_{ref} is defined in a two-part manner (i.e. as room temperature for the layer to be printed and as output from the trained model for the printed layer(s)) due to the layer-by-layer nature of DED. More details on the complications from it are presented in Section 2.2.2.

The values of the material and process parameters introduced in this section are listed in Appendix B.

2.2. PINN setup

2.2.1. 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:

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

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

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

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 [27]. 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. Superscript $\square^*$ indicates non-dimensionalised quantities:

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

The following characteristic length, time, and temperature are adopted:

$$\begin{cases} l_c &= \phi_{\text{laser}} \\ t_c &= \frac{\phi_{\text{laser}}}{v_{\text{laser}}} \\ T_c &= 3000 \end{cases} \quad (11)$$

where ϕ_{laser} is the diameter of the laser spot and v_{laser} is the speed of the laser movement. T_c is set as 3000K to contain any possible extreme over-prediction from the melting point of the material (Ti6Al4V).

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

$$\tilde{\mathcal{F}}(\hat{T}^*, \mathbf{x}^*, t^*) = \frac{\partial \hat{T}^*}{\partial t^*} - \frac{k}{\rho C_p} \frac{t_c}{l_c^2} \left(\frac{\partial^2 \hat{T}^*}{\partial x^{*2}} + \frac{\partial^2 \hat{T}^*}{\partial y^{*2}} \right) - \frac{1}{\rho C_p} \frac{t_c}{T_c} \frac{6\eta \bar{P}}{\pi ab} e^{-3 \left(\frac{(x^* - (x_0^* + v^* t^*))^2}{a^2} + \frac{(y^* - y_{\text{laser}}^*)^2}{b^2} \right)} \quad (12)$$

$$\tilde{\mathcal{B}}(\hat{T}^*, \mathbf{x}^*, t^*) = \begin{cases} \pm \frac{\partial \hat{T}^*}{\partial n^*} - \frac{h}{k} l_c (\hat{T}^* - T_{\text{amb}}^*) - \frac{\sigma \epsilon}{k} T_c^3 l_c (\hat{T}^{*4} - T_{\text{amb}}^{*4}) \\ \frac{\partial \hat{T}^*}{\partial y^*} - \frac{h_{\text{force}}}{k} l_c (\hat{T}^* - T_{\text{amb}}^*) \end{cases} \quad (13)$$

$$\tilde{\mathcal{I}}(\hat{T}^*, \mathbf{x}^*, 0) = \hat{T}_{t=t_0}^* - T_{\text{ref}}^* \quad (14)$$

Normalisation of -1 to 1 is applied to the inputs when the collocation points are fed into the NN:

$$\begin{cases} \mathbf{x}_{\text{in}}^* &= 2 \frac{\mathbf{x}^* - \mathbf{x}_{\text{min}}^*}{\mathbf{x}_{\text{max}}^* - \mathbf{x}_{\text{min}}^*} - 1 \\ t_{\text{in}}^* &= 2 \frac{t^* - t_{\text{min}}^*}{t_{\text{max}}^* - t_{\text{min}}^*} - 1 \end{cases} \quad (15)$$

where subscript $\square_{\text{in}}$ indicate the value fed into the hidden layers of the NN, $\mathbf{x}_{\text{max}}^*, \mathbf{x}_{\text{min}}^*, t_{\text{max}}^*, t_{\text{min}}^*$ are the upper and lower bounds of the non-dimensionalised spatial and temporal domain.

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

$$\mathcal{L}_{\text{PDE}} = \frac{1}{N_{\text{PDE}}} \sum_{i=1}^{N_{\text{PDE}}} \left| \tilde{\mathcal{F}}(\hat{T}^*(\mathbf{x}_{\text{PDE}}^{*i}, t_{\text{PDE}}^{*i}), \mathbf{x}_{\text{PDE}}^{*i}, t_{\text{PDE}}^{*i}) \right|^2 \quad (16)$$

$$\mathcal{L}_{\text{BC}} = \frac{1}{N_{\text{BC}}} \sum_{i=1}^{N_{\text{BC}}} \left| \tilde{\mathcal{B}}(\hat{T}^*(\mathbf{x}_{\text{BC}}^{*i}, t_{\text{BC}}^{*i}), \mathbf{x}_{\text{BC}}^{*i}, t_{\text{BC}}^{*i}) \right|^2 \quad (17)$$

$$\mathcal{L}_{\text{IC}} = \frac{1}{N_{\text{IC}}} \sum_{i=1}^{N_{\text{IC}}} \left| \tilde{\mathcal{I}}(\hat{T}^*(\mathbf{x}_{\text{IC}}^{*i}, 0), \mathbf{x}_{\text{IC}}^{*i}, 0, w_{\text{ic}}^i) \right|^2 \quad (18)$$

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 18 is the pointwise weight for the initial condition loss which is defined in Equation 22 and a more detailed explanation of this term is provided in Section 2.3.2.

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

$$\mathcal{L} = \frac{w_{\text{PDE}} \mathcal{L}_{\text{PDE}} + w_{\text{BC}} \mathcal{L}_{\text{BC}} + w_{\text{IC}} \mathcal{L}_{\text{IC}}}{w_{\text{PDE}} + w_{\text{BC}} + w_{\text{IC}}} \quad (19)$$

The loss expressions are embedded into that NN architecture as illustrated in Figure 1.

2.2.2. Spatio-temporal sampling points

Collocation points, akin to those in numerical methods, are spatiotemporal sampling points used during the PINN training for computing the relevant loss terms to evaluate the level of approximation. When PINN is applied as a forward solver without any auxiliary data, the collocation points play an especially 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 [24, 28, 29]. The weighted total loss in Equation 19 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 the 2D implementation of 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 $\alpha_s \phi_{\text{laser}}$ over each time step. In this work, α_s is chosen to be 0.15 for sufficient overlapping.

Figure 3 illustrates the four main zones into which the domain can be split: 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.

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 $(4\phi_{\text{laser}} \times 1.5\phi_{\text{laser}})$ 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 3. 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 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. The IC collocation points are put into two sets – one for the printed part (Zone A and Substrate) and one for the new layer (Zone B and 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

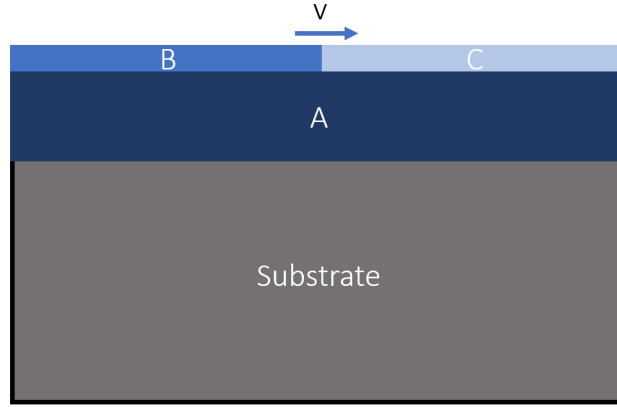


Figure 3: Key zones during the multi-layer process: Zone A, B, and C constitute the part region. Zone A is the already printed layer. Zone B and C constitute the layer that is being printed – Zone B has material deposited and Zone C is where the material will be deposited as indicated by the velocity vector (v) of the heat source.

challenges to the training. However, it is necessary given the additive nature of the DED process. The challenge and the solution will be discussed in more detail in Section 2.3.2.

Figure 4 illustrates the distribution of the generated collocation points for the first layer in printing over the print duration.

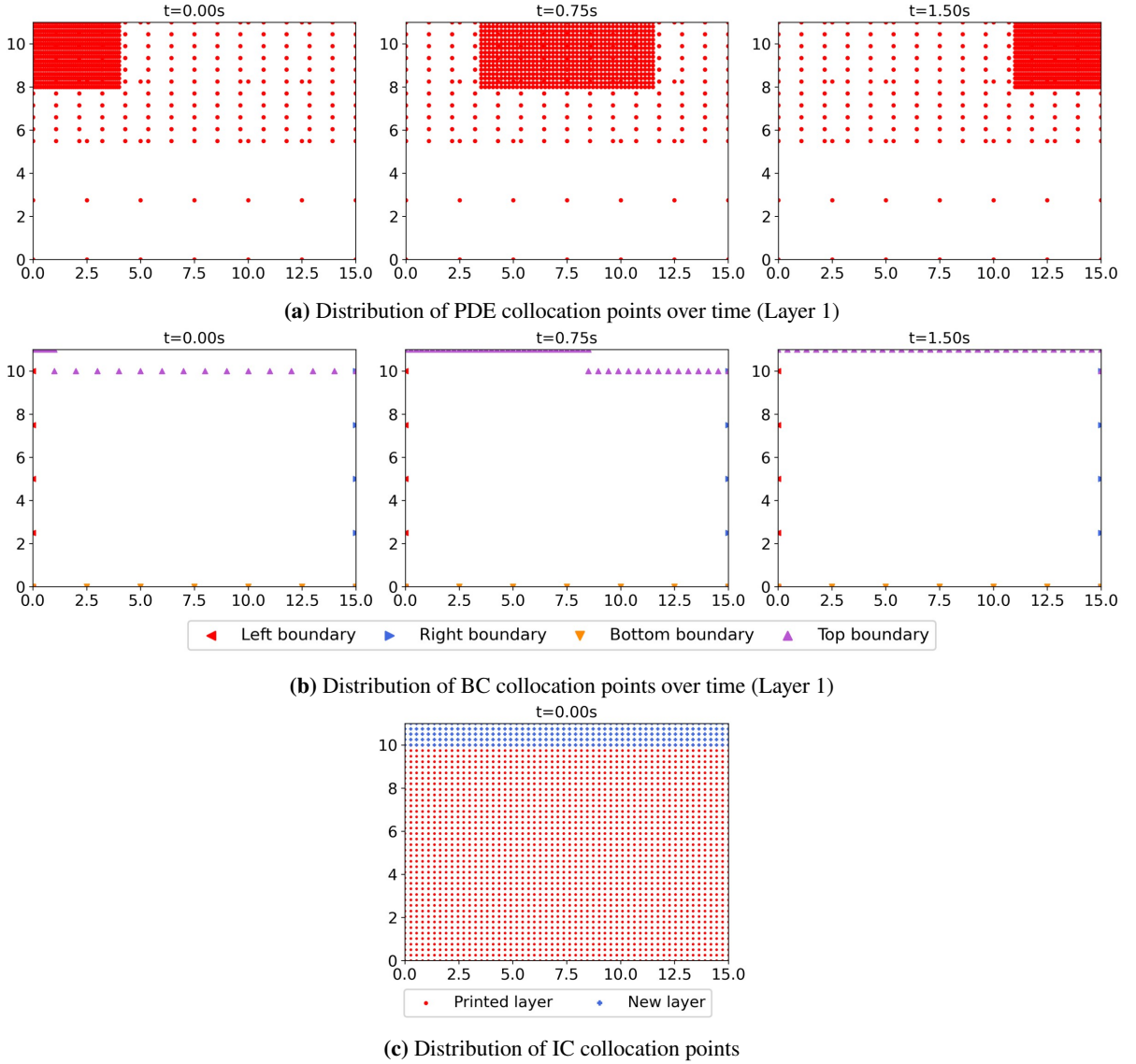


Figure 4: Distribution of the collocation points for the first layer (Layer 1). The x-axis and y-axis correspond to the length and height of the part domain in a Cartesian coordinate system. The bottom left corner of the part domain is defined as the global origin, indicated by (0,0) and the values are nominal with a unit of *mm*.

. The convention is used for all subsequent figures where the part domain is presented.

It is reported in the literature that randomised allocation of the collocation points in the given spatial and temporal domain may result in better generality [19, 28]. No significant difference is observed herein when the two strategies are applied. Hence, grid-based collocation allocation is used in the 2D implementations for better reproducibility, and random collocation allocation is used in 3D for easier and more precise control of the number and density of collocation points herein. The number of the collocation points should be determined through a convergence study in which the number of points is increased until the correct feature/solution is captured with acceptable accuracy. In this work, ANSYS simulation result is taken as the benchmark and the number of collocation points is deemed sufficient when the RMSE is below 100K.

The values of the bespoke hyperparameters introduced in this section are listed in Appendix B.

2.3. Multi-layer 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 cases investigated in this article (i.e. continuous rectangular plate), each layer consists of one scan, assuming that there is no change of direction of the scan within the layer. Figure 5 illustrates the workflow of the PINN-based thermal simulation for the multi-layer problem.

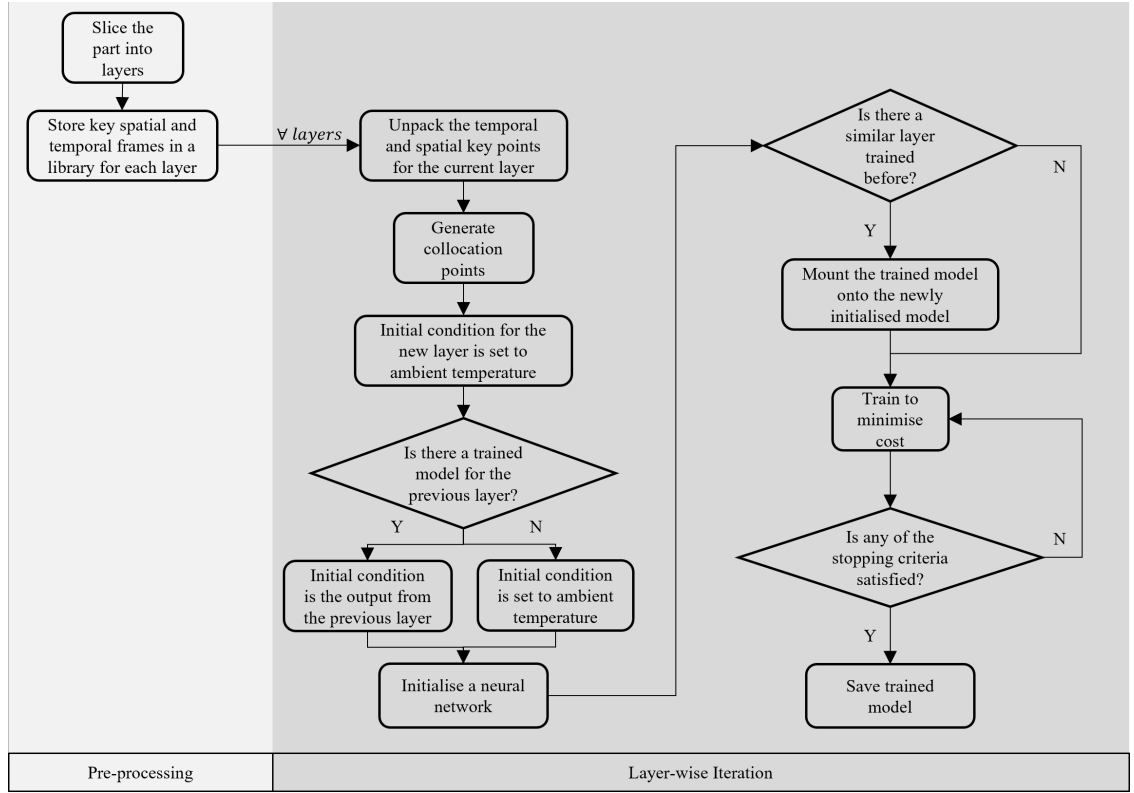


Figure 5: Flowchart illustrating the overall framework

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 layer. 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. A ‘similar scan path’ is defined as one having the same scanning direction. Such initialisation from a pre-trained model is termed ‘warm-start initialisation’ in this paper. The training in each iteration will stop when either the maximum epoch number or the threshold loss/causality value is met.

The challenge of using PINN for the multi-layer problem is the conflict between the discontinuous nature intrinsic to the DED process and NN's deficiency in approximating discontinuity. To overcome it, strategies that approximate the discontinuities are devised and presented in the following section.

2.3.1. Laser on-off switch

The on-off state of the laser is dictated by the temporal marker(s) during each scan which is 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 20 and the laser switch function $laser_switch(t)$ is expressed as Equation 21.

$$\dot{e}_{gen}^* = laser_switch(t^*) \frac{1}{\rho C_p} \frac{t_c}{T_c} \frac{6\eta \bar{P}}{\pi ab} e^{-3 \left(\frac{(x^* - (x_0^* + v^* t^*))^2}{a^2} + \frac{(y^* - y_{laser}^*)^2}{b^2} \right)} \quad (20)$$

$$laser_switch(t^*) = \sum_{i=1}^N \left(\frac{1}{1 + e^{-\alpha_i(t^* - t_{start_i}^*)}} - \frac{1}{1 + e^{-\alpha_i(t^* - t_{stop_i}^*)}} \right) \quad (21)$$

where N is the total instances of switching on the laser (i.e. $N \geq 1$). Parameter α_i 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, after systematically adjusting the values such that the RMSE against ANSYS is no more than 100K. A similar process is applied to determine the values of other bespoke hyperparameters presented in the later sections.

2.3.2. Pointwise weight assignment for IC losses

With the additive nature, a new layer is activated and initialised at the beginning of each scan. It implies that the initial condition assignment for all layers except for the first layer needs to have two components: 1) the layers in Zone A and Substrate take the output from the previously trained model at the instance when the new layer starts printing as the initial condition; 2) the new layer (Zone B and C) takes the ambient temperature as the initial condition. The two components are illustrated in Figure 4c as the red points and the green points, respectively. Figure 6 illustrates the erroneous output from the trained model when the later component of the IC is not specified – the temperature at the start of the new layer should be the ambient temperature since no material has been deposited yet.

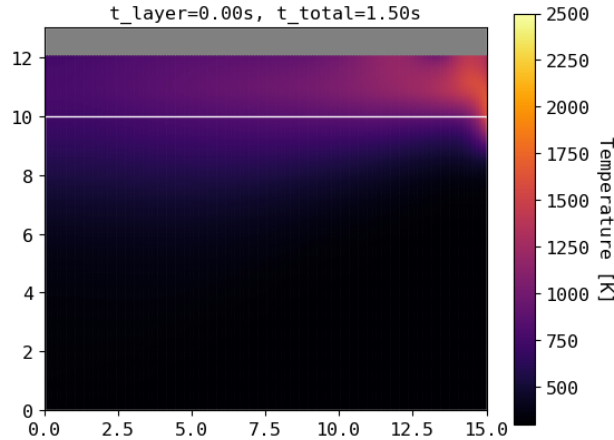


Figure 6: Temperature output when IC is not specified for the new layer (figure showing the beginning of the 2nd layer): the region of height 11-12mm should be room temperature as no material is yet deposited.

Following a preliminary investigation on the error values at the IC collocation points over epochs, it can be concluded that the large IC loss values are contributed by the collocation points close to the interface between the new layer and the printed part (i.e. the discontinuity). Hence, a method to assign a weight for each collocation point is implemented. The points closer to the interface are assigned a much lower weight and the overall IC loss is a weighted sum of the points. It prevents completely losing the information closer to the interface but also reduces the magnitude

of the error values which smoothens the loss topology and allows the training to continue. The weight for each point is determined through Equation 22:

$$w_{ic}^i = \frac{1}{e^{-\alpha_w(|y_i - y_{\text{interface}}| - \beta h_{\text{layer}})}} \quad (22)$$

where h_{layer} is the layer thickness, α_w and β are parameters controlling how fast and where the pointwise weight starts to decay to 0, respectively. In this study, α_w and β are assigned values of 200 and 0.05, respectively.

2.3.3. 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 and B) and void (Zone C) is achieved through the use of sigmoid function and can be expressed as:

$$\begin{aligned} \tilde{F} = & \tilde{F}^{\text{solid}} \left(\left[\left(1 + e^{\alpha_z \Delta x_b^*} \right) \left(1 + e^{-\alpha_z \Delta y_b^*} \right) \right]^{-1} + \left[1 + e^{\alpha_z \Delta y_b^*} \right]^{-1} \right) \\ & + \tilde{F}^{\text{air}} \left[\left(1 + e^{-\alpha_z \Delta x_b^*} \right) \left(1 + e^{-\alpha_z \Delta y_b^*} \right) \right]^{-1} \end{aligned} \quad (23)$$

where

$$\begin{cases} \Delta x_b^* &= x^* - x_{\text{boundary}}^* \\ \Delta y_b^* &= y^* - y_{\text{boundary}}^* \end{cases}$$

and 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.

2.3.4. Training with causality

Incorporating causality in the training is proposed in [30] 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:

$$w_c^i = e^{-\epsilon \left(\sum_{k=1}^{i-1} \mathcal{L}_t(t_k^*, \mathbf{x}^*) + w_{IC} \mathcal{L}_{IC} \right)} \quad \text{for } i = 2, 3, \dots, N_t \quad (24)$$

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:

$$\mathcal{L}_t^i = w_{\text{PDE}} \mathcal{L}_{\text{PDE}}(t^{*i}, \mathbf{x}_{\text{PDE}}^{*i}) + w_{\text{BC}} \mathcal{L}_{\text{BC}}(t^{*i}, \mathbf{x}_{\text{BC}}^{*i}) \quad \text{for } i = 2, 3, \dots, N_t \quad (25)$$

The causal weight prevents a timestep from contributing to the overall loss until the loss from all previous timesteps converges to a sufficiently small value.

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

$$\mathcal{L} = \frac{1}{N_t} \sum_{i=0}^{N_t} \left(\frac{w_c^i}{w_{\text{PDE}} + w_{\text{BC}} + w_{\text{IC}}} \mathcal{L}_t^i + \frac{w_{\text{IC}}}{w_{\text{PDE}} + w_{\text{BC}} + w_{\text{IC}}} \mathcal{L}_{\text{IC}} \right) \quad (26)$$

2.3.5. Adaptive weight for loss terms

In [27], the author proposed a ‘self-adaptive global weight’ algorithm which adjusts the weights for the $\mathcal{L}_{\text{PDE}}$, $\mathcal{L}_{\text{BC}}$, and $\mathcal{L}_{\text{IC}}$ according to the following equations:

$$\hat{\lambda}_{\text{PDE}} = \frac{\|\nabla_{\theta} \mathcal{L}_{\text{PDE}}(\theta)\| + \|\nabla_{\theta} \mathcal{L}_{\text{BC}}(\theta)\| + \|\nabla_{\theta} \mathcal{L}_{\text{IC}}(\theta)\|}{\|\nabla_{\theta} \mathcal{L}_{\text{PDE}}(\theta)\|} \quad (27)$$

$$\hat{\lambda}_{\text{BC}} = \frac{\|\nabla_{\theta} \mathcal{L}_{\text{PDE}}(\theta)\| + \|\nabla_{\theta} \mathcal{L}_{\text{BC}}(\theta)\| + \|\nabla_{\theta} \mathcal{L}_{\text{IC}}(\theta)\|}{\|\nabla_{\theta} \mathcal{L}_{\text{BC}}(\theta)\|} \quad (28)$$

$$\hat{\lambda}_{\text{IC}} = \frac{\|\nabla_{\theta} \mathcal{L}_{\text{PDE}}(\theta)\| + \|\nabla_{\theta} \mathcal{L}_{\text{BC}}(\theta)\| + \|\nabla_{\theta} \mathcal{L}_{\text{IC}}(\theta)\|}{\|\nabla_{\theta} \mathcal{L}_{\text{IC}}(\theta)\|} \quad (29)$$

The new weights are computed and obtained as a rolling average as the current ones every set epochs:

$$\lambda_{\text{new}} = \alpha \lambda_{\text{old}} + (1 - \alpha) \hat{\lambda}_{\text{new}} \quad (30)$$

Such a scheme claims to balance the potentially large difference in orders of magnitude among different losses during training and hence avoid biasing towards any loss term during training. The effectiveness of such scheme is investigated in this work.

2.4. Numerical benchmark

The numerical benchmark is generated by running a detailed thermal simulation in ANSYS (ANSYS Mechanical ADPL 2024). The same implementation has been adopted for calibration in MIS-related works by Prof. Albert To's group [10, 11, 12, 13, 14]. The same boundary conditions and Goldak heat source model as entailed in Section 2.1 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.

In the benchmark comparison, 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. The resolution is chosen following a mesh convergence study and the details of it are presented in the Appendix.

2.5. Case studies

To evaluate the effectiveness of the proposed strategies, three case studies based on a simple three-layer case (15mm×3mm) are conducted – 1) pointwise weight allocation for IC, 2) explicit solid-void definition, and 3) alternative stopping criterion. The metrics used for comparison include the loss evolution, the output accuracy based on benchmark error, and other training-related parameters. Due to the multi-objective optimisation nature of PINN training, the choice of weights will affect the loss landscape (a space of all loss values corresponding to all parameter combinations) and influence the final result which is essentially finding the minima (ideally the global minimum) in it. To ensure fair evaluation, a suitable set of weights is first determined from a grid search and utilised for the rest of the case studies.

Determining loss weights: Four configurations outlined in Table 1 are assessed. The first configuration of equal weight forms the baseline for the comparison. The second and the third configurations are chosen based on the assumption that the satisfaction of PDE and IC is more vital for accurate output, with the former being more extreme in enforcing such an assumption. Finally, the fourth configuration is applied to investigate if the satisfaction of IC is more critical for obtaining accurate prediction.

Table 1: Combinations of loss weights being tested

Configuration #	w_{PDE}	w_{BC}	w_{IC}
I	1	1	1
II	10	1	10
III	2	1	2
IV	1	1	5

Pointwise weight allocation for IC: The performance of the proposed pointwise IC approach is compared against the alternatives, specifically, simple assignment of the two-part IC and offsetting from the interface.

Explicit solid-void definitions: The performance of training with the proposed explicit solid-void definition is compared against that without.

Alternative stopping criteria: In the recent paper introducing causality into the PINN training process, the author suggests that causal weight w_c^i can act as an alternative stopping criterion [30]. A 2-layer case is run with $\min(w_c^i) \geq 0.98$ as the stopping criterion and a maximum epoch of 200,000 is imposed to prevent infinite running. The performance is compared against the same case trained with the conventional early stopping criterion where a threshold total loss value is used.

The framework is implemented in Python and the PINN is constructed through Pytorch [31]. More details on the parameters are presented in Appendix B.

3. Results and Discussion

3.1. Case studies

3.1.1. Determine the loss weights

Three factors, 1) loss values, 2) output accuracy, and 3) compatibility with causal weight, are used to evaluate and select the most suitable loss weight configurations.

If only the loss values were assessed, Configuration IV would be an ideal choice as it has the lowest magnitude in both nominal loss values (Figure 7a) which indicate the degree of agreement for the PDE, BC, and IC at the end of training and the weighted counterparts (Figure 7b) which illustrate the relative importance of the loss terms in training.

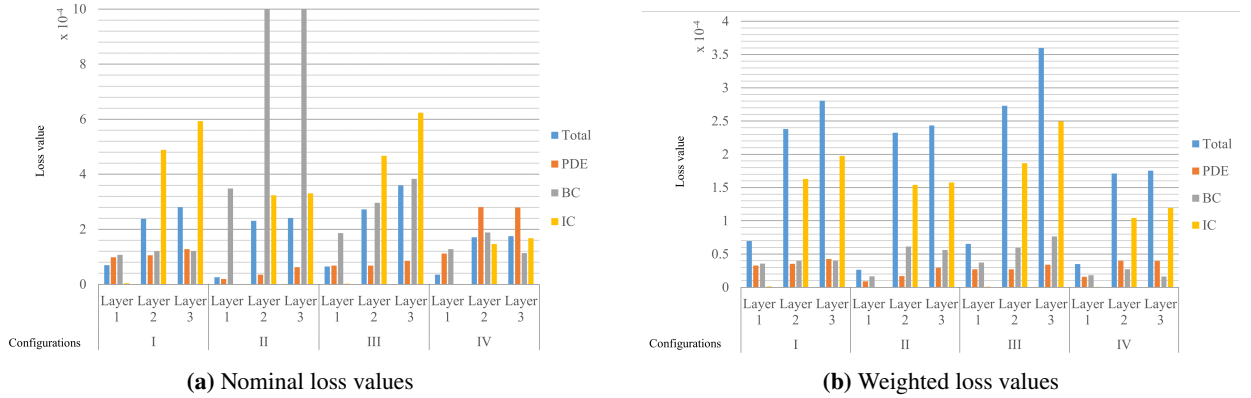


Figure 7: Loss values at the end of 100,000 epochs for the weight configurations

Conversely, upon inspecting the error values compared to the benchmark from ANSYS, Configuration IV results in the worst accuracy, especially in the third layer. It could be attributed to the much higher nominal PDE loss compared to other configurations which supports the hypothesis that both PDE and IC play vital roles in improving the accuracy of the trained model. Consequently, Configuration II appears to be a better option purely from the accuracy point of view. Nevertheless, further inspection of Figure 8 indicates a much higher spread of causal weights, translating to a reduced level of satisfaction of causality. It also implies that a lower threshold value or more epochs are needed for testing stopping with causal weight compared to other configurations should causal weight be used as an alternative stopping criterion.

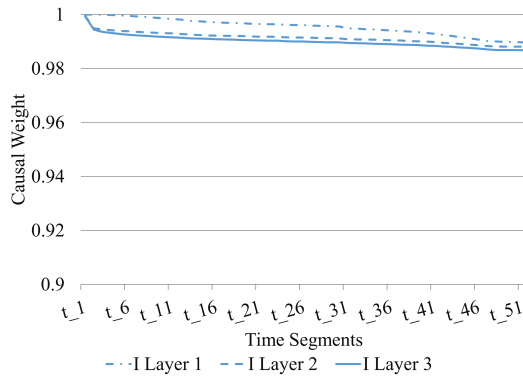
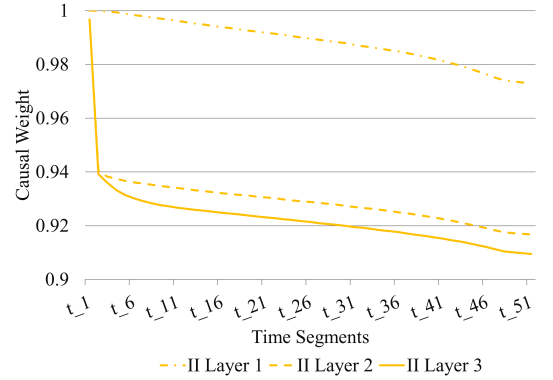
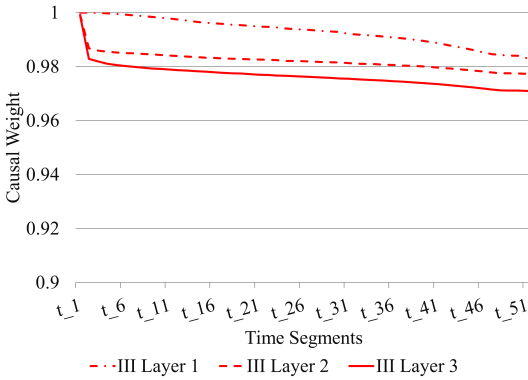
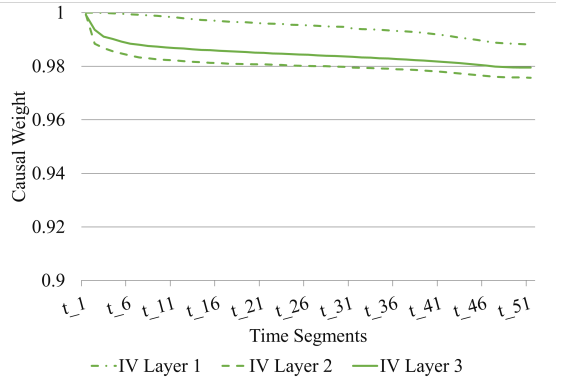
Considering all the metrics holistically, Configuration III is chosen as the most suitable. It should be acknowledged that optimising the choice of weights is an active area of research, similar to other hyperparameter tunings, and obtaining the optimal weights is beyond the scope of this work.

3.1.2. Evaluation on pointwise weight assignment for IC

By simply assigning the two-part IC, difficulty in the convergence of the IC loss and total loss is observed (Figure 9). The observation agrees with the consensus on NN's deficiency in approximating discontinuity. The stagnated total loss value, often used as a stopping criterion for training, poses challenges for the model to effectively traverse the loss landscape and may lead it to stuck in local minima. Therefore, it necessitates techniques to mitigate the issue.

Table 2: Root Mean Square Error (RMSE) and Maximum Absolute Error (Max. Abs.) compared to ANSYS benchmark result (best values highlighted in bold).

Configuration #	Layer 1		Layer 2		Layer 3	
	RMSE [K]	Max. Abs. [K]	RMSE [K]	Max. Abs. [K]	RMSE [K]	Max. Abs. [K]
I	21.75	424	32.15	334	48.15	318
II	18.74	366	35.21	221	52.63	322
III	20.48	419	41.31	307	47.88	230
IV	21.98	428.9	34.66	332.03	106.92	458.32

**(a) Weight configuration I****(b) Weight configuration II****(c) Weight configuration III****(d) Weight configuration IV****Figure 8:** Causal weights at the end of 100,000 epochs for all weight configurations

An intuitive and straightforward approach to tackle the large errors stagnating at the interface between the old and new layers over training is to offset the collocation points away from the interface. However, as evidenced by the RMSE values in Table 3, it results in a detrimental effect on the model's accuracy – less information is made available in one of the most critical regions to capture the dynamics. It is further supported by the observation that the RMSE from offsetting is at a similar order of magnitude as that without two-part IC considered – no information is fed in either case. Moreover, examining the loss evolution in Figure 10a and 10c reveals that the IC loss still stagnates and is only marginally lower than that without offsetting. The difficulty in lowering the IC loss and total loss (often used as the stopping criterion) persists.

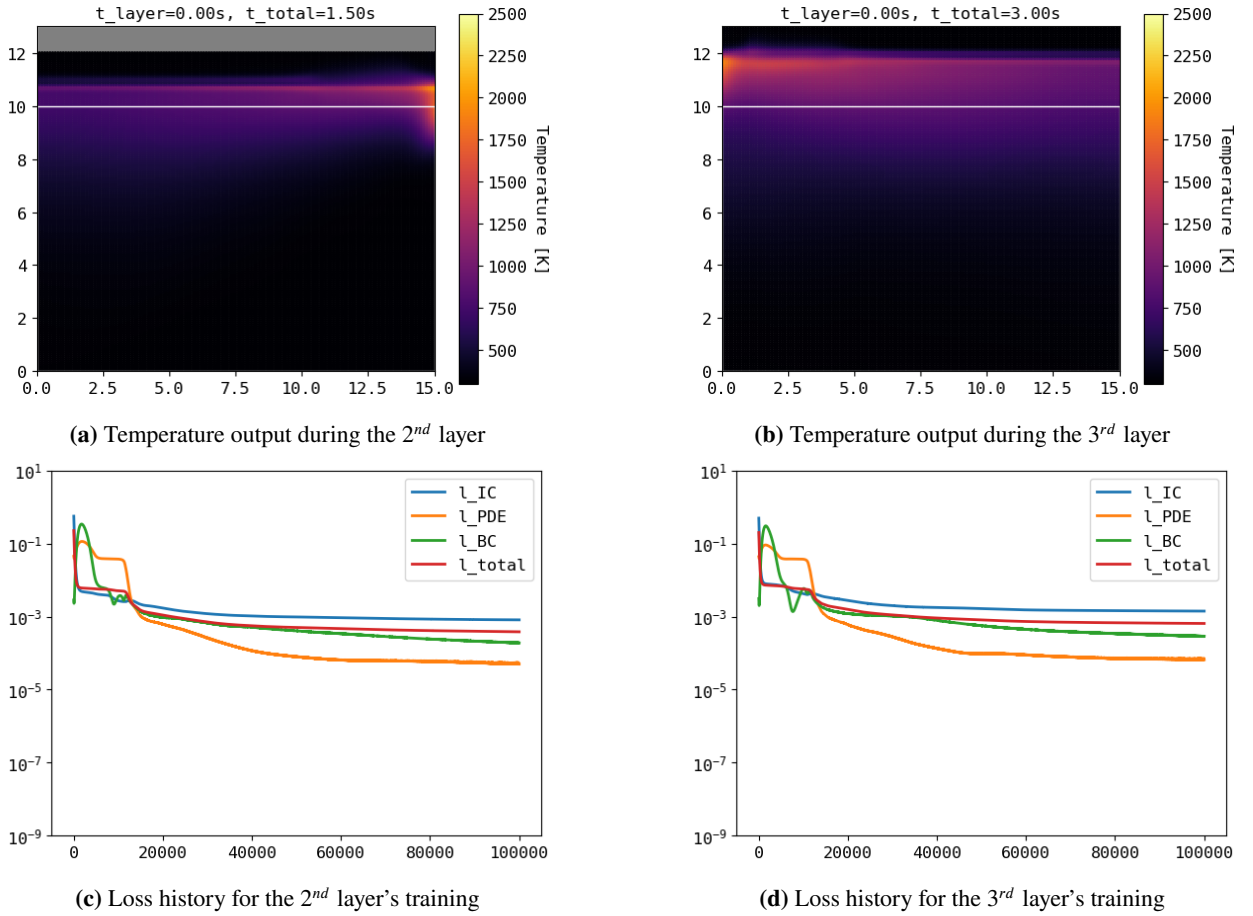
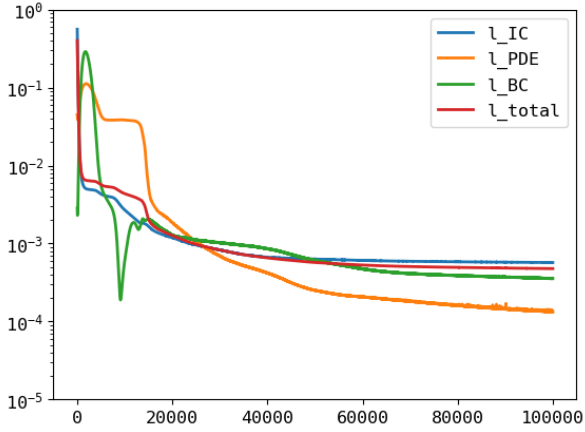


Figure 9: Outputs and loss history with simple assignment of IC for the new layer

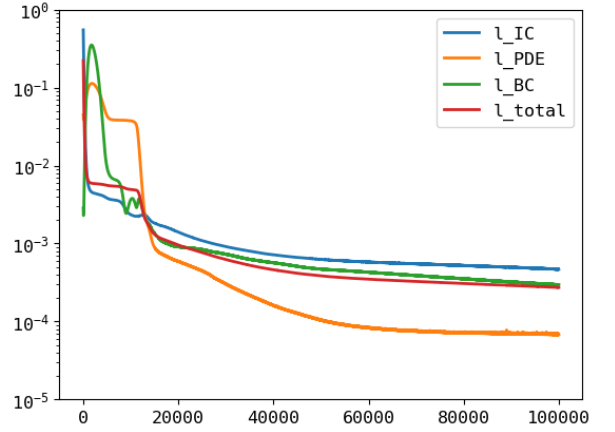
Table 3: Root Mean Square Error (RMSE) and Maximum Absolute Error (Max. Abs.) compared to ANSYS benchmark result for different IC strategies.

Case	Layer 1		Layer 2		Layer 3	
	RMSE [K]	Max. Abs. [K]	RMSE [K]	Max. Abs. [K]	RMSE [K]	Max. Abs. [K]
No two-part IC	22.71	414.07	53.92	278.39	108.22	447.32
Simply apply two-part IC	22.16	433.79	62.42	483.70	86.31	397.93
Offset for two-part IC	22.32	422.09	51.11	416.19	151.90	611.91
Pointwise Weight for two-part IC	20.48	419.29	41.31	307.37	47.88	229.68

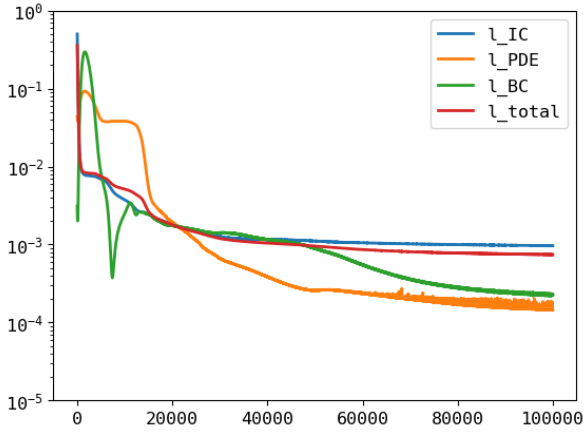
Unlike the simple offset approach where information is completely lost, the pointwise weight assignment numerically attenuates the contribution from the abnormally high error values at the interface. It implies that the model is made aware of the interface during training without the stagnated loss. The expected benefit in convergence



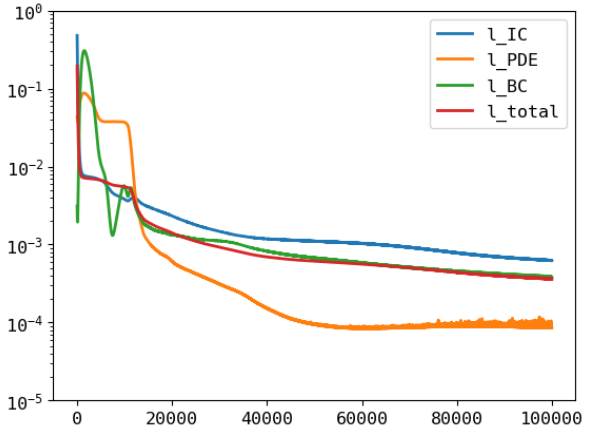
(a) Loss history for the 2nd layer's training using simple offset method



(b) Loss history for the 2nd layer's training using pointwise weight method



(c) Loss history for the 3rd layer's training using simple offset method



(d) Loss history for the 3rd layer's training using pointwise weight method

Figure 10: Loss histories comparison between offsetting the IC collocation points $0.05 * h_{layer}$ from the interface and applying pointwise weight for IC losses. Compared to those from the pointwise weight method, the IC and total losses stagnate with the simple offset method and a higher PDE loss at the end of the training is also observed.

is observed from the continually decreasing IC and total loss in Figure 10. The much lower RMSEs across multiple layers in Table 3 indicate the pointwise weight's effectiveness in balancing the increased awareness of discontinuity and not stagnating the numerical loss values for training convergence.

3.1.3. Evaluation on explicit solid-void definition

Figure 11 illustrates the rapid decrease in temperature at the deposited layer (i.e. Zone A) when the sigmoid activation of metal and void zone is not applied. It agrees with the consensus that the model will maintain the already learnt features when no additional information is made available. With the pointwise weight in place, the trained model ‘learns’ that the new layer (i.e. Zone B and Zone C) takes an initial room temperature while the existing part has a residual temperature distribution as the output from the previous training. When no explicit definition of the void region is assigned (Figure 11), the trained model applies the heat transfer property of the solid while keeping the temperature in Zone C as the room temperature, essentially an infinite heat sink on top of the previously deposited layer. It explains the rapid temperature loss, making it necessary to assign the heat transfer properties to at least part of Zone C to model the correct amount of heat loss.

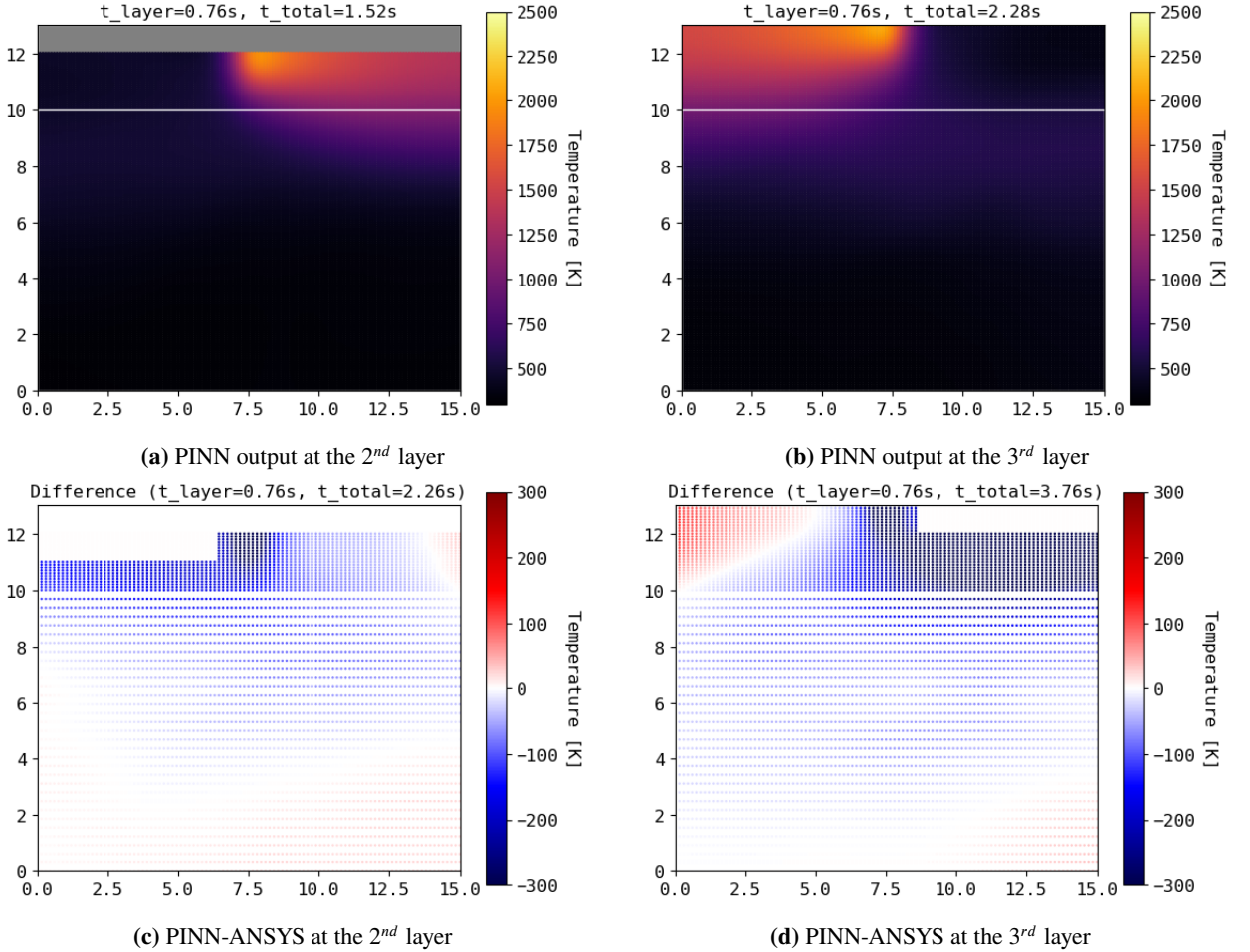


Figure 11: Significant underprediction of temperature at the deposited layer when the thermal properties of air are not explicitly modelled

Using a sigmoid function to activate the solid or void region approximates the transition between regions as a smooth function. It prevents introducing another moving discontinuity for approximation which would increase training difficulty. Since the training is collocation-point-based, the sigmoid-based smooth transition is only effective if the collocation points close to the discontinuity are dense – otherwise, the information that is available to the model is equivalent to having a discontinuity. It is another driver for the denser set of collocation points that move with the heat source and the material boundary.

Besides the much lower RMSE values as listed in Table 4, the effectiveness of the solid-void transition is also illustrated by comparing the spatial derivatives for the cases without and with the explicit definition of solid-void

properties – a clear boundary of temperature change with increasing height (i.e. $\frac{\partial T}{\partial y}$) is observed at the boundary between Zone A and C for the case where solid-void properties are clearly defined (Figure 12). It supports that the model was able to learn the different thermal properties.

Table 4: Root Mean Square Error (RMSE) and Maximum Absolute Error (Max. Abs.) compared to ANSYS benchmark result with and without solid-void regions clearly defined.

Case	Layer 1		Layer 2		Layer 3	
	RMSE [K]	Max. Abs. [K]	RMSE [K]	Max. Abs. [K]	RMSE [K]	Max. Abs. [K]
No solid-void	19.78	453.67	57.18	430.32	105.75	599.90
With solid-void	20.48	419.29	41.31	307.37	47.88	229.68

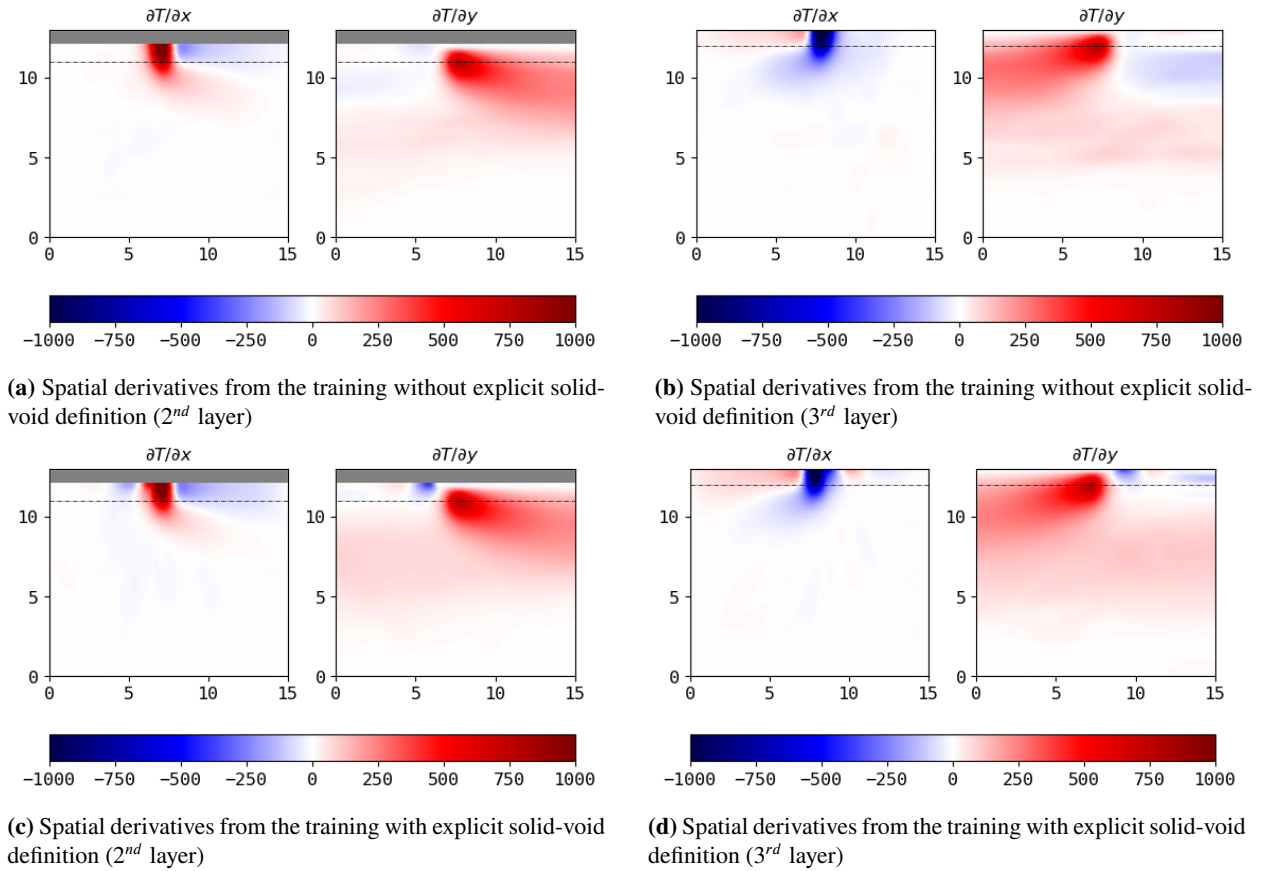


Figure 12: Comparison of spatial derivatives for cases with and without solid-void material properties explicitly defined.

Figure 13 presents a sample raw output from the trained model and its comparison against the ANSYS benchmark result as well as the post-processed output and the corresponding comparison. The latter comparison is used in this paper. Figure 13a and 13b highlight a limitation in the proposed framework – while the temperature of the solid region is predicted with reasonable accuracy presented in the later section, the temperature of the void region is likely not predicted accurately. While it is unlikely that the region above the current layer is the uniform room temperature as a result of ANSYS' prediction based on the element activation method, it is probably not as high as that predicted by the

trained model. Since the region of interest is the printed part (i.e. solid region), there is little information (or a need for information) in the void region as long as accurate prediction is achieved in the actual area of interest. With no concrete conclusion on the temperature in the void region from the benchmark methods, a post-processing that adjusts the temperature to room temperature is applied. It ensures fairness of comparison since the region is not (and does not need to be) simulated accurately. It is more pertinent to compare if the correct heat retention is predicted by the trained model and Figure 13d indicates so, suggesting that the model is able to ‘learn’ the change from solid to void heat transfer properties with the smoothened solid-void activation.

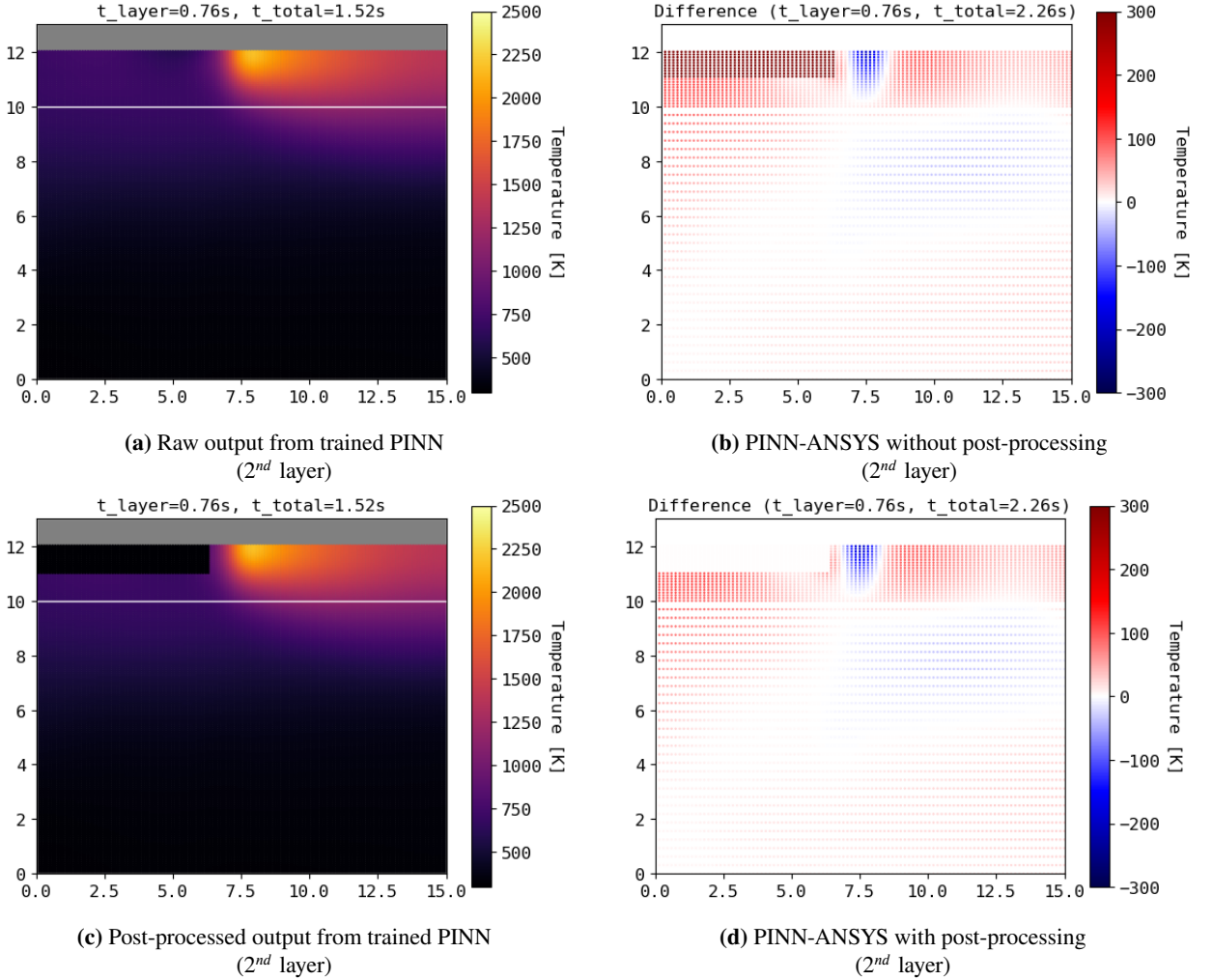


Figure 13: Raw output from trained PINN model and the effect of post-processing

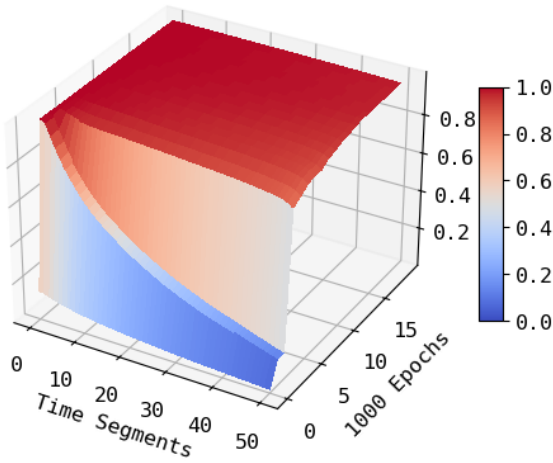
3.1.4. Effect of using alternative stopping criterion

A stronger positive correlation between high causal weights and better output accuracy is indicated by the comparison in Table 5, suggesting the causal weight to be a better stopping criterion than an arbitrary total loss threshold. It is expected considering the temporal nature of DED process. Conversely, it is also observed that the training with causal weight as the stopping criteria might be longer than with total loss when the initial condition is more challenging. Figure 14 illustrates that the improvement of fitting for the later time segments is plateauing. It indicates that the model is facing challenges learning the later time segments which could be due to the expressibility of the selected PINN architecture.

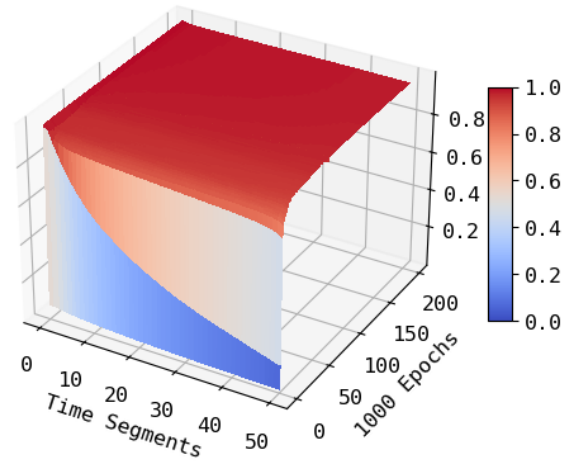
Table 5 also revealed that the randomness in training could lead to instability and fluctuation of final output when cost value is used as the stopping criterion as shown by the very high benchmark error for the second layer despite having the same configuration as the 3-layer case run to generate the results. However, stopping with a causal weight limit seems to reduce the fluctuation of the output. Nevertheless, further and more comprehensive investigation is deemed necessary for other means to mitigate the fluctuation in training, ideally with theoretical backing.

Table 5: Performance comparison between the stopping criteria

	Layer 1		Layer 2	
	Stopping with Loss Value	Stopping with Causal Weight	Stopping with Loss Value	Stopping with Causal Weight
$\mathcal{L}_{\text{PDE}}$	1.11e-4	1.77e-4	2.74e-4	2.64e-4
$\mathcal{L}_{\text{BC}}$	1.27e-4	2.43e-4	1.23e-4	1.54e-4
$\mathcal{L}_{\text{IC}}$	1.13e-6	2.06e-6	1.20e-4	1.39e-4
$\mathcal{L}_{\text{total}}$	3.48e-5	6.09e-5	1.42e-4	1.58e-4
Epochs	100000	19881	100000	200000
RMSE [K]	27.4	26.3	133.9	45.7



(a) Causal weight evolution over epochs (1st layer)



(b) Causal weight evolution over epochs (2nd layer)

Figure 14: Evolution of causal weight over epochs.

3.2. Prediction from trained model

With the loss weights chosen as $\{w_{PDE} = 2, w_{BC} = 1, w_{IC} = 2\}$ and applying all pointwise weight for IC as well as explicit solid-void thermal property definition, the PINN model is trained and loss history and sample outputs are presented in Figure 15. The diffusion behaviour of the heat and a reasonable general shape of the heat source is captured by the trained model. The accumulated heat from previous layers is also represented well. For a more quantitative evaluation of the quality of the output, the benchmark comparison is presented in the following section.

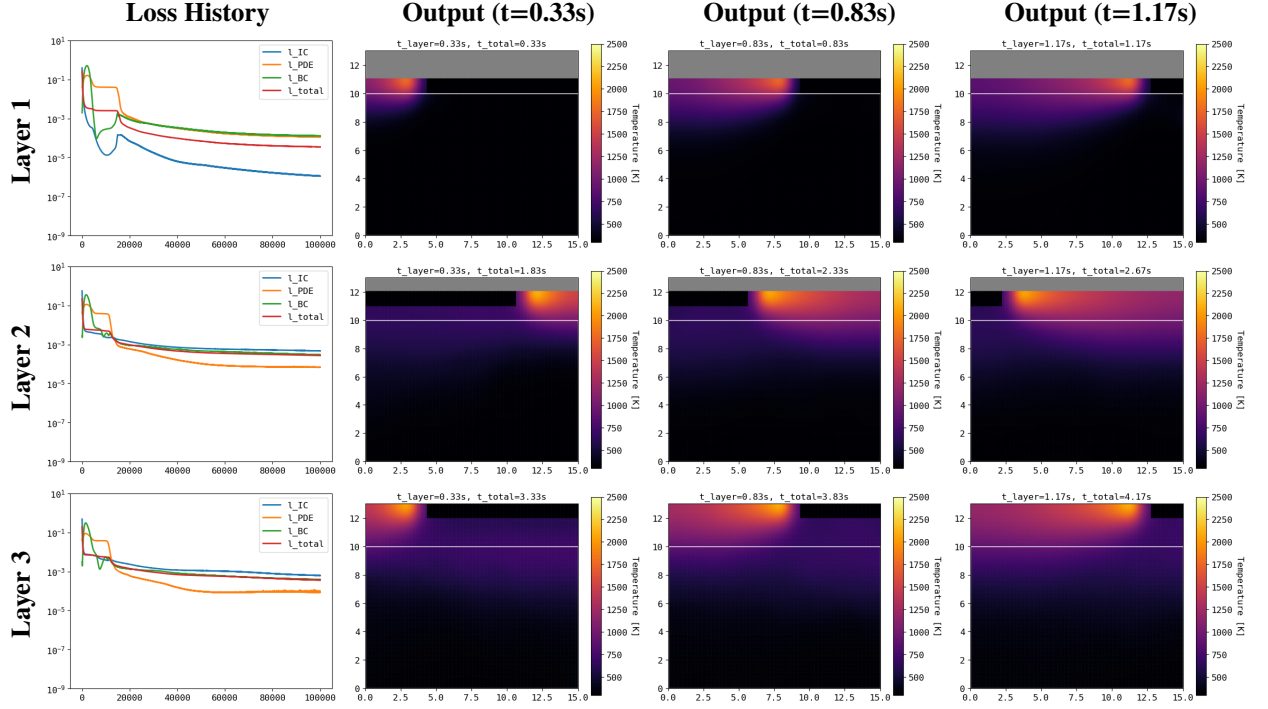


Figure 15: Loss history and temperature prediction for the 15mm×3mm case. For each layer, the outputs at $t=0.33$, 0.83 and 1.17 s are presented.

3.2.1. Numerical benchmark

Quantitative comparisons between the PINN results and the benchmark results from ANSYS are made to validate the accuracy of the model. Root mean square errors of 20.48K, 41.31K, and 47.88K are observed for the 1st, 2nd, and 3rd layers, respectively and Figure 16 illustrates the distribution of errors. The highest magnitude is observed at the core of the heat source for both layers where the PINN result under-predicts the peak temperature by about 10%. Furthermore, over-prediction behind the heat source is observed for all layers, indicating less heat is dissipated to the surroundings than predicted by the ANSYS simulation.

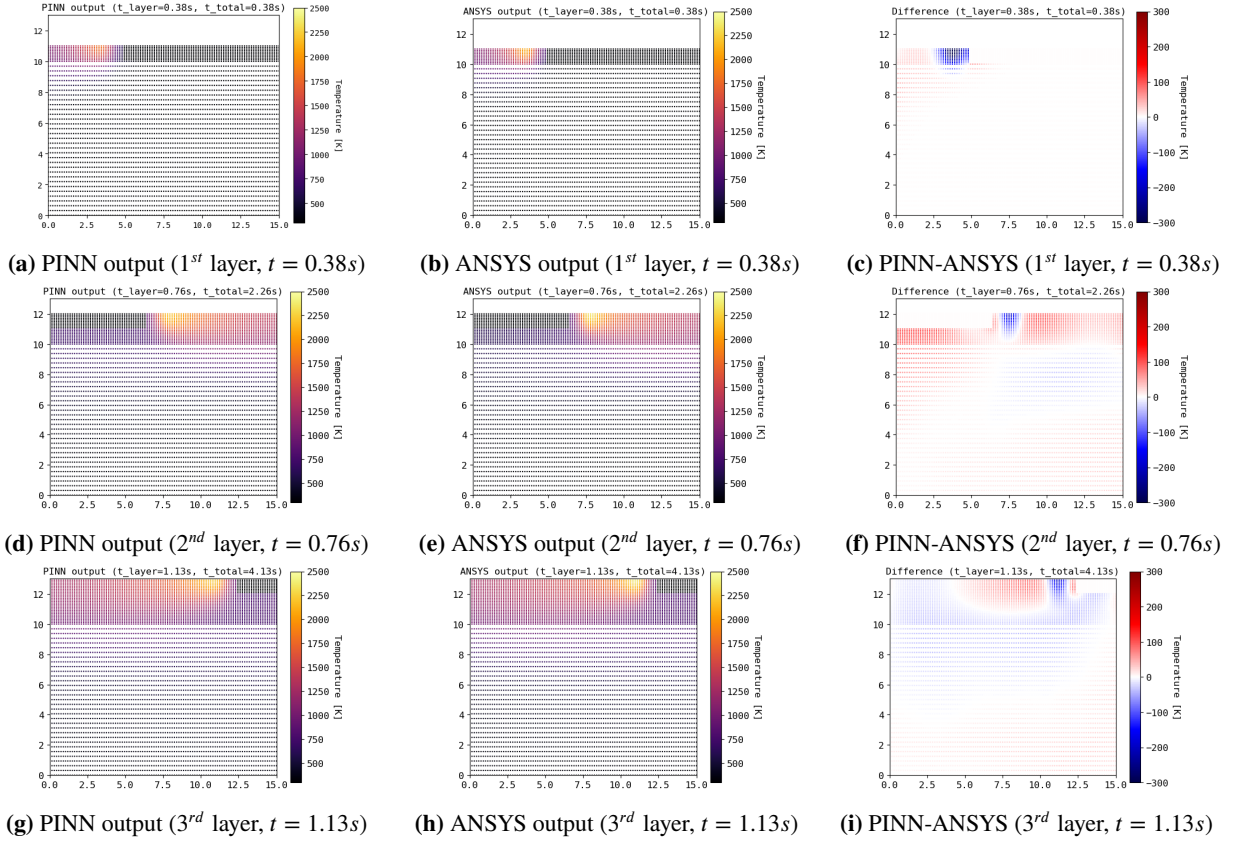
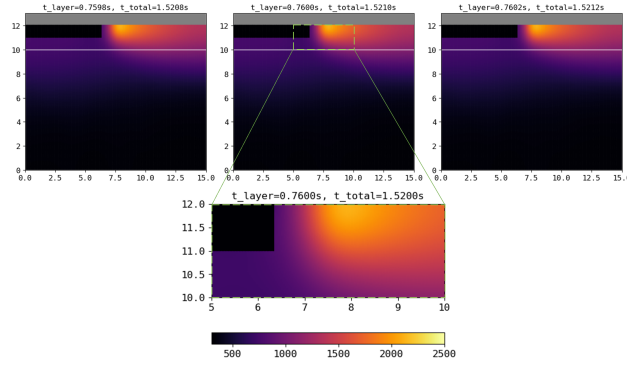


Figure 16: Benchmark comparison between the PINN and ANSYS simulations. A root mean square error of 20.48K, 41.31K, and 47.88K is observed for the 1st, 2nd, and 3rd layers, respectively.

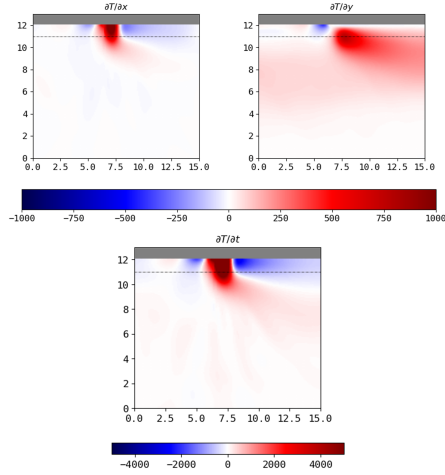
The under-prediction of the core temperature compared to ANSYS benchmark result reveals a key artefact of the NN-based solution – the extrema are less sharply defined due to training cost being defined as weighted mean values from all collocation points. Hence, the extrema are smoothened in the approximation. Nevertheless, the underprediction is limited to within 10% in magnitude and within close proximity of the melt pool. The overprediction behind the heat source is likely attributed to the under-representation of BCs (with the smallest number of collocation points) coupled with the low weights assigned to BC in training since PDE and IC are identified to play a more critical role in the multi-layer framework. It can be argued that the highest error occurs close to the boundary where even the accuracy of the benchmark result is not guaranteed. Nevertheless, more investigation will be pursued to alleviate the error. One potential method is applying hard constraints for the BCs in PINN architecture [32].

3.3. Super-resolution capability and availability of derivative information

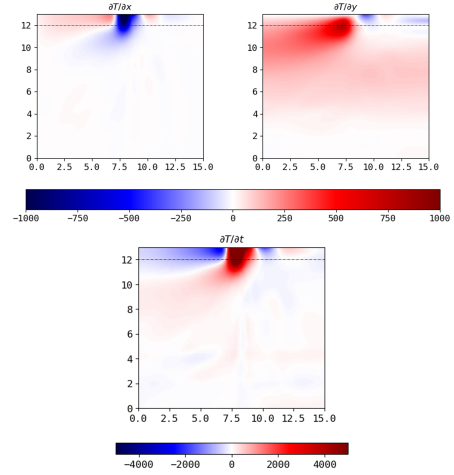
Unlike the ANSYS solution, the training PINN model's solution is meshless, i.e. the solution is not limited by the temporal nor spatial discretisation during the solving/training process. Such capability is illustrated in Figure 17a where temporal-super-resolution and spatial-super-resolution of the solution are conducted and no artefact is observed. It implies two potential benefits – 1) the training time could be further reduced by adopting a coarser discretisation for collocation point assignment; 2) investigation at a much finer spatial and/or temporal scale can be conducted which is critical for a highly transient problem such as during DED or other metal AM processes.



(a) Temporal super-resolution of 144 times and spatial super-resolution of 100 times of those used during training. (Layer 2)



(b) Temperature gradients and cooling rate. (Layer 2)



(c) Temperature gradients and cooling rate. (Layer 3)

Figure 17: Demonstration of spatiotemporal super-resolution and availability of derivative information from the trained PINN model.

In addition to the benefits of the meshless characteristics, the PINN solution also produces the derivatives at no additional cost – the temperature gradients and cooling rates at any point can be obtained in the same manner as the temperature output with the same meshless characteristics. The derivative information has been utilised and stored in the gradient graph during the training process. Hence, unlike those from the numerical solution which is limited by the mesh resolution and can only be obtained after the whole temperature field is solved, the derivative information from PINN can be obtained during the evaluation on-demand. Figure 17b and 17c illustrate sample outputs of the gradient information across layers. The positive value of the temperature gradient is defined as a temperature rise in the positive spatial direction, resulting in the flip of the sign of the $\partial T/\partial x$ values in Figure 17b since the heat source is moving from right to left. The greatest temperature gradient is observed near the centre of the heat source, consistent with the expectation. A band of positive $\partial T/\partial y$ is observed with an increase in band height from Figure 17b to 17c, corresponding to the heat accumulation and downward diffusion as layers build up. The quickest temperature

rise and drop are observed at the centre and immediately behind the heat source, respectively. While most of the observations agree with expectations, artefacts are nevertheless present. Island features are detected in the $\partial T/\partial x$ and $\partial T/\partial t$ plots. They are likely due to incomplete training (i.e. the errors are not reduced to 0 during training) and potentially the grid distribution of collocation points. The magnitude of the artefact is relatively low and is unlikely to pose a significant adverse impact on the output. Nevertheless, future efforts will be made to reduce the presence of such artefacts. Recalling that the thermally induced failures during metal AM are often attributed to the rapid cooling and large thermal gradient, a solution based on PINN could potentially allow direct and quick prediction residual stress/strain prediction based on the actual thermal gradient and cooling rate information instead of waiting for the full temperature field to be solved or relying on calibration. It would encourage additional theoretical research correlating the thermal gradient and cooling rate with residual stress/strain.

3.4. Potential time saving

To demonstrate the time-saving capability of the proposed framework, a larger, 10-layer part (15mm×10mm) is simulated by both training the PINN and ANSYS simulation. Figure 18 illustrates the comparison of time required for both methods. Taking advantage of initialisation with prior knowledge – i.e. ‘warm-start’ initialisation, the training time for the later layers is substantially reduced once a similar process is trained for. In the current implementation, a minimum number of 5000 epochs is imposed before training can end early when $\min(w_c^l) \geq 0.98$, resulting in the plateauing of computational time reduction in the later layers as observed.

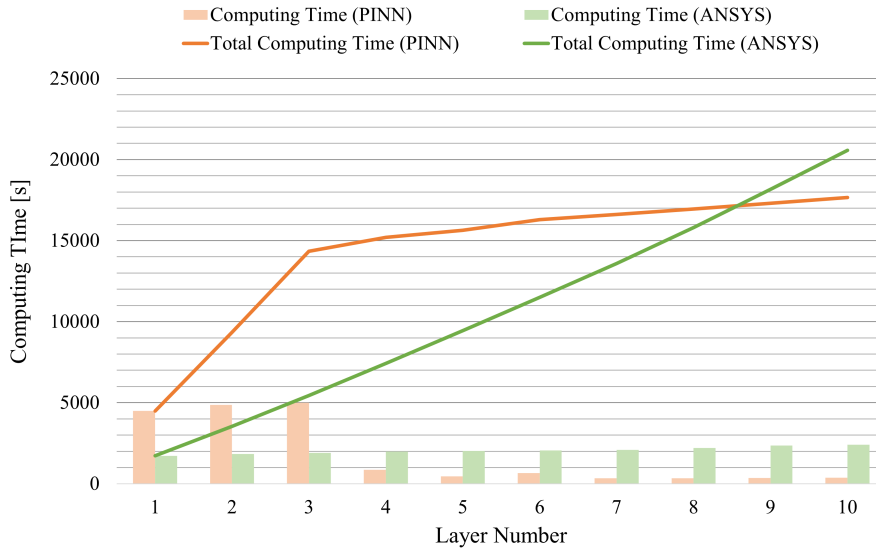


Figure 18: Computational time comparison between PINN and ANSYS.

While a scan of the same direction works well as a criterion to define a ‘similar scan’, it remains an open question to quantify how much similarity in configuration is sufficient for warm-start to work. It should also be acknowledged that the same definition of a ‘similar’ layer based on scan direction might no longer be valid for more complex profiles. Other criteria defining a ‘similar scan’ will be investigated in further detail in the future.

3.5. Extension from 2D to 3D

The extension from 2D to 3D involves minimal effort in modifying the governing equations and the loss functions as detailed in Appendix A. Figure 19 illustrates the output from the trained model for the 3D case with two layers printed (i.e. 15mm×2mm×2mm) on a substrate of 15mm×2mm×10mm.

With the increased dimension, collocation points are sampled through a random distribution instead of a uniform grid for better control of the number and/or density of the collocation points. As indicated in Table 6, the density of collocation points for the 3D case is less than the equivalent volumetric density from the 2D cases while yielding a similar level of agreement with the ANSYS prediction. The lower collocation point density in 3D was primarily

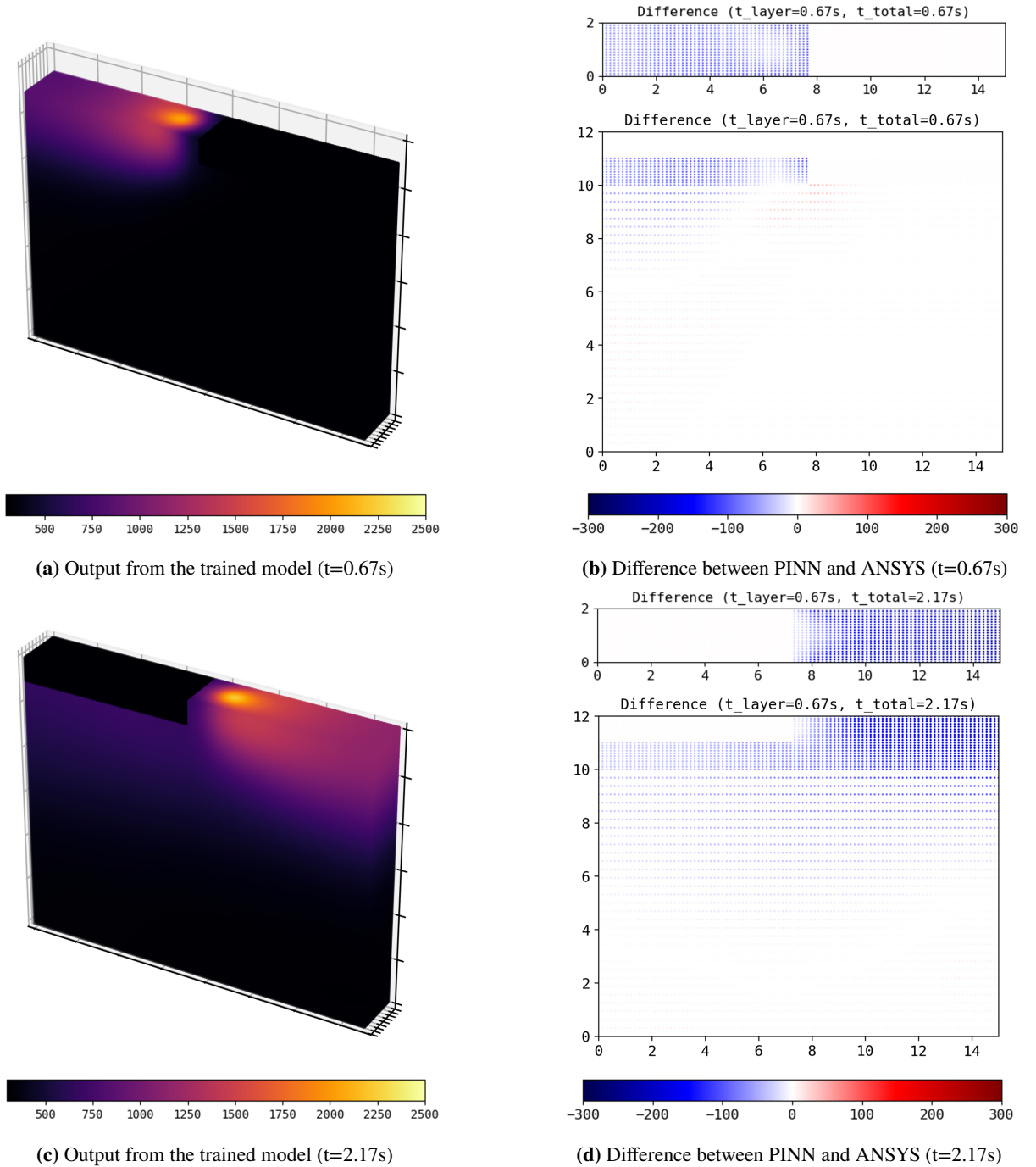


Figure 19: Prediction from the trained PINN model for the 3D case and difference between the PINN and ANSYS results on the top face ($z=11\text{mm}$ and 12mm) and the middle plane ($y=1\text{mm}$). All temperature values are in K.

driven by controlling the increase in computational time. An additional 2D test was run based on the 2D equivalent planar density based on the 3D implementation to verify that the reduction in density does not result in a significant reduction in prediction quality. An RMSE of 16.77K was reported. Such observation indicates that further reduction of the collocation points can potentially be achieved and the determination of the minimum collocation density as part of the future hyper-parameter tuning should be carried out in the future.

Table 6: Density of collocation points and the associated RMSE in 2D and 3D cases

		2D			3D	
		Number of collocation points	Planer/edge density	Equivalent volume/face density	Number of collocation points	Volume/face density
PDE	Global	35	0.212	0.0977	330	1.00
	Secondary	165	2.20	3.26	330	2.20
	Local	704	29.3	159	1727	72.0
	sub-total	904	5.48	12.8	2387	7.23
BC sub-total		67	1.34	1.80	400	1.00
IC sub-total		2520	15.3	59.7	2640	8.00
RMSE [K]			20.48			19.42

3.6. Recommendations on using PINN for DED temperature history prediction

The applicability of the strategies to improve PINN training in the literature [27, 33] is not always universal across applications. Remarks on the effectiveness of some commonly cited ones are made, and insights on improving the performance of PINN as a forward solver for predicting the DED temperature history are offered herein.

3.6.1. Effectiveness of general recommendations regarding PINN training for DED temperature history prediction

The thermal history simulation of DED process is characterised by 1) high localised temperature, 2) high-temperature gradient, 3) discontinuous temperature field as an initial condition at the beginning of a new scan, and 4) being dynamic.

The first two characteristics necessitate normalisation and more importantly, non-dimensionalisation of the loss functions to prevent saturated gradient during training. It agrees with the recommendation in [27]. There are, however, works in the literature [24] that used the nominal values during training and obtained plausible results for single-layer temperature history prediction for DED. Nevertheless, it has been observed that non-dimensionalisation and normalisation significantly stabilised the training process and facilitated convergence – depending on the configuration, using nominal values may fail to train altogether.

Training with causality as proposed in [30] allows temporal causality in the problem to be learnt during the training process which improves the quality of prediction in a dynamic problem such as temperature history prediction for DED. The improvement is pronounced, especially for the later layers where there is a discontinuity in the initial condition. A prerequisite for causality to work is non-dimensionalisation and normalisation. Training with causality using nominal values in DED temperature history prediction will not converge.

Adaptive weight for the loss terms is proposed in [27] and is claimed to prevent bias towards any loss terms when there is a large discrepancy of magnitudes. In this work, the benefit of the method is only observed to a limited extent – the loss values usually reduce to a similar order of magnitude within 10,000 epochs (e.g. Figure 10) when a fixed weight assignment is applied. In the 3D implementation, applying adaptive weight from the start of the training could even prevent PDE loss from being trained due to the larger magnitude of error upon initialisation. On the other hand, the stabilising effect of adaptive weight is observed if it is applied after training for 20,000 epochs, returning more consistent results.

Techniques such as modified multi-layer perceptron (modified MLP) and Fourier feature embedding proposed in [27] do not lead to observable improvements to the training or the result. Due to the relatively shallow layers, applying modified MLP significantly increases the computing time due to the additional two layers.

3.6.2. Prediction improvement through intermittent injection of data

One deficiency observed in the 3D implementation is the deteriorated agreement with ANSYS prediction when the problem is solved as a pure forward problem over multiple layers as illustrated in Table 7. It is mainly attributed to the accumulated under-prediction as time progresses – the PINN prediction of the heat source core temperature tends to be an underprediction due to the difficulty in capturing the highly non-linear, singular value at the core. It results in a lower predicted global temperature which is set as the starting temperature distribution for the next scan, leading to

a cumulative effect over layers. Such observation is more prominent in 3D due to the increased amount of collocation points across the domain – the loss value used for training is a weighted average of all collocation points.

Table 7: Root-mean-square error for two-layer scan without and with injecting ANSYS results at the end of Layer 1 into the training process.

		Without Injection of Data	With Injection of Data
RMSE [K]	Layer 1	23.43	28.31
	Layer 2	92.94	44.79

Such an issue can be alleviated through the injection of data which clears the cumulated under-prediction over the last layer(s). The ability to incorporate data offers potential continuous improvement of the model's prediction. The data injected are the temperature of printed layers when the new layer is initialised (i.e. initial condition). They take the form of target temperature for the IC collocation points in the already printed layer (Zone A and Substrate). For demonstration purpose, 2880 points are randomly sampled from the numerical simulation at the end of the previous layer. In eventuality where the framework is applied to experiments, the points are more likely to be from experimental observation instead (e.g. via in-situ monitoring). The trained model remains a predictor (Class 1 of the forward problem) even with the injection of data since they are restricted to the initial condition based on the past result and no data from the temporal duration to be modelled is used.

The intermittent inclusion of data for the prediction of temperature history during the DED process is the recommended means of using the proposed framework – unlike numerical approaches, PINN models can easily take in experiential observations and use the data where possible to improve the agreement between predictions and experimental observation. The ability to combine imperfect physics and limited data is where the true strength of PINN lies.

4. Conclusion

In this paper, we presented a framework that allows multi-layer thermal history prediction for DED to be achieved based on PINN. Benefiting from strategies that make NN contiguous with the discontinuities that are intrinsic to the DED process, i.e. the pointwise weight assignment for IC loss and explicit definition of the solid-void regions, the proposed framework is able to achieve a similar level of accuracy as the numerical methods with the potential of significant time-saving for large-scale parts.

Such a framework sets the foundation for employing PINN in the thermomechanical simulation for practical DED applications. Instead of acting as a replacement for conventional numerical methods, PINN's unique capability of incorporating both physical laws and experimental data implies that the proposed framework can either achieve continuous improvement in prediction when combined with data (e.g. from in-situ monitoring) as a forward solver, or be applied to the inverse problem where effective thermal properties are generalised from experimental observations, allowing computationally cheap, conduction-based numerical models to be utilised while maintaining good agreement with observations. Moreover, PINN models not only generate temperature history but also thermal gradients and cooling rates. Such information can be of use for alternative mechanical models that predict the thermally induced distortions on the fly. It opens up the possibility of optimisation with regard to thermal distortion which is an extremely exciting field of research.

Future endeavours based on the presented framework can be outlined as follows. While the proposed strategies provided remedies that overcome the shortfall of NN in approximating discontinuities, the accuracy and ease of training can be further improved. Besides the standard hyper-parameters of ML training, the proposed strategies introduce bespoke hyper-parameters. The tuning of such parameters, a non-trivial task, will likely affect the training process and hence the optimality of the performance of the trained model. In addition to the hyperparameter tuning to find the most optimal model complexity and training configurations, the application of alternative architectures should be explored. Techniques strengthening robustness against discontinuity and enhancing generality are of particular interest. Theoretical work on reducing the fluctuation and randomness in the training outcome will greatly benefit the ML-based methods, including the framework proposed in this paper, instilling confidence in applying them to real-life applications. Material property determination based on solving the inverse problem through PINN with experimental data will also be explored, especially in combination with in-situ monitoring.

Acknowledgements This work is funded and supported by the Department of Aeronautics at Imperial College London. The authors would like to thank Prof. Albert To and Dr Wen Dong for the ANSYS APDL scripts used to generate the benchmark results, and Dr János Plocher (ANSYS Germany GmbH) for the advice on setting up the ANSYS simulation.

Data Availability The PINN codes used during the current study are available from the corresponding author upon reasonable request.

CRedit authorship contribution statement

Bohan Peng: Writing – Original draft, Validation, Methodology, Investigation, Data curation, Conceptualization, Software. **Ajit Panesar:** Writing – Review & Editing, Supervision, Resources, Funding acquisition, Conceptualization.

References

- [1] Saurabh Vaidya, Prashant Ambad, and Santosh Bhosle. Industry 4.0 – A Glimpse. *Procedia Manufacturing*, 20:233–238, January 2018. ISSN 2351-9789. doi: 10.1016/j.promfg.2018.02.034. URL <https://www.sciencedirect.com/science/article/pii/S2351978918300672>.
- [2] William E. Frazier. Metal Additive Manufacturing: A Review. *Journal of Materials Engineering and Performance*, 23(6):1917–1928, June 2014. ISSN 1544-1024. doi: 10.1007/s11665-014-0958-z. URL <https://doi.org/10.1007/s11665-014-0958-z>.
- [3] Monsuru Ramoni, Ragavanantham Shanmugam, N. Thangapandian, and M. Vishnuvarthanan. Challenges in Additive Manufacturing for Metals and Alloys. In M. Adam Khan and J. T. Winowlin Jappes, editors, *Innovations in Additive Manufacturing*, Springer Tracts in Additive Manufacturing, pages 57–72. Springer International Publishing, Cham, 2022. ISBN 978-3-030-89401-6. doi: 10.1007/978-3-030-89401-6_3. URL https://doi.org/10.1007/978-3-030-89401-6_3.

- [4] Mohamad Bayat, Wen Dong, Jesper Thorborg, Albert C. To, and Jesper H. Hattel. A review of multi-scale and multi-physics simulations of metal additive manufacturing processes with focus on modeling strategies. *Additive Manufacturing*, 47:102278, November 2021. ISSN 22148604. doi: 10.1016/j.addma.2021.102278. URL <https://linkinghub.elsevier.com/retrieve/pii/S2214860421004383>.
- [5] Erik R. Denlinger, Michael Gouge, Jeff Irwin, and Pan Michaleris. Thermomechanical model development and in situ experimental validation of the Laser Powder-Bed Fusion process. *Additive Manufacturing*, 16:73–80, August 2017. ISSN 2214-8604. doi: 10.1016/j.addma.2017.05.001. URL <https://www.sciencedirect.com/science/article/pii/S2214860417300441>.
- [6] Alaa Olleak and Zhimin Xi. A scan-wise adaptive remeshing framework for thermal simulation of the selective laser melting process. *The International Journal of Advanced Manufacturing Technology*, 107(1):573–584, March 2020. ISSN 1433-3015. doi: 10.1007/s00170-020-04995-7. URL <https://doi.org/10.1007/s00170-020-04995-7>.
- [7] T. P. Moran, D. H. Warner, and N. Phan. Scan-by-scan part-scale thermal modelling for defect prediction in metal additive manufacturing. *Additive Manufacturing*, 37:101667, January 2021. ISSN 2214-8604. doi: 10.1016/j.addma.2020.101667. URL <https://www.sciencedirect.com/science/article/pii/S2214860420310393>.
- [8] L. Parry, I. A. Ashcroft, and R. D. Wildman. Understanding the effect of laser scan strategy on residual stress in selective laser melting through thermo-mechanical simulation. *Additive Manufacturing*, 12:1–15, October 2016. ISSN 2214-8604. doi: 10.1016/j.addma.2016.05.014. URL <https://www.sciencedirect.com/science/article/pii/S2214860416300987>.
- [9] Bo Cheng, Subin Shrestha, and Kevin Chou. Stress and deformation evaluations of scanning strategy effect in selective laser melting. *Additive Manufacturing*, 12:240–251, October 2016. ISSN 2214-8604. doi: 10.1016/j.addma.2016.05.007. URL <https://www.sciencedirect.com/science/article/pii/S2214860416300914>.
- [10] Xuan Liang, Lin Cheng, Qian Chen, Qingcheng Yang, and Albert C. To. 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*, 23:471–486, October 2018. ISSN 22148604. doi: 10.1016/j.addma.2018.08.029. URL <https://linkinghub.elsevier.com/retrieve/pii/S2214860418304858>.
- [11] Qian Chen, Xuan Liang, Devlin Hayduke, Jikai Liu, Lin Cheng, Jason Oskin, Ryan Whitmore, and Albert C. To. An inherent strain based multiscale modeling framework for simulating part-scale residual deformation for direct metal laser sintering. *Additive Manufacturing*, 28:406–418, August 2019. ISSN 22148604. doi: 10.1016/j.addma.2019.05.021. URL <https://linkinghub.elsevier.com/retrieve/pii/S2214860418310583>.
- [12] Wen Dong, Xuan Liang, Qian Chen, Shawn Hinnebusch, Zekai Zhou, and Albert C. To. A new procedure for implementing the modified inherent strain method with improved accuracy in predicting both residual stress and deformation for laser powder bed fusion. *Additive Manufacturing*, 47:102345, November 2021. ISSN 22148604. doi: 10.1016/j.addma.2021.102345. URL <https://linkinghub.elsevier.com/retrieve/pii/S2214860421005030>.
- [13] Xuan Liang, Wen Dong, Qian Chen, and Albert C. To. On incorporating scanning strategy effects into the modified inherent strain modeling framework for laser powder bed fusion. *Additive Manufacturing*, 37:101648, January 2021. ISSN 22148604. doi: 10.1016/j.addma.2020.101648. URL <https://linkinghub.elsevier.com/retrieve/pii/S2214860420310204>.
- [14] Wen Dong, Xavier A. Jimenez, and Albert C. To. Temperature-dependent modified inherent strain method for predicting residual stress and distortion of Ti6Al4V walls manufactured by wire-arc directed energy deposition. *Additive Manufacturing*, 62:103386, January 2023. ISSN 22148604. doi: 10.1016/j.addma.2022.103386. URL <https://linkinghub.elsevier.com/retrieve/pii/S2214860422007758>.
- [15] Kahraman Demir, Zhizhou Zhang, Adi Ben-Artzy, Peter Hosemann, and Grace X. Gu. Laser scan strategy descriptor for defect prognosis in metal additive manufacturing using neural networks. *Journal of Manufacturing Processes*, 67:628–634, July 2021. ISSN 1526-6125. doi: 10.1016/j.jmapro.2021.05.011. URL <https://www.sciencedirect.com/science/article/pii/S1526612521003339>.
- [16] K. Ren, Y. Chew, Y. F. Zhang, J. Y. H. Fuh, and G. J. Bi. Thermal field prediction for laser scanning paths in laser aided additive manufacturing by physics-based machine learning. *Computer Methods in Applied Mechanics and Engineering*, 362:112734, April 2020. ISSN 0045-7825. doi: 10.1016/j.cma.2019.112734. URL <https://www.sciencedirect.com/science/article/pii/S0045782519306243>.
- [17] Ruochen Hong, Lei Zhang, Joseph Lifton, Stephen Daynes, Jun Wei, Stefanie Feih, and Wen Feng Lu. Artificial neural network-based geometry compensation to improve the printing accuracy of selective laser melting fabricated sub-millimetre overhang trusses. *Additive Manufacturing*, 37:101594, January 2021. ISSN 2214-8604. doi: 10.1016/j.addma.2020.101594. URL <https://www.sciencedirect.com/science/article/pii/S2214860420309660>.
- [18] Francis Ogoke, William Lee, Ning-Yu Kao, Alexander Myers, Jack Beuth, Jonathan Malen, and Amir Barati Farimani. Convolutional neural networks for melt depth prediction and visualization in laser powder bed fusion. *The International Journal of Advanced Manufacturing Technology*, October 2023. ISSN 1433-3015. doi: 10.1007/s00170-023-12384-z. URL <https://doi.org/10.1007/s00170-023-12384-z>.
- [19] M. Raissi, P. Perdikaris, and G. E. Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707, 2 2019. ISSN 0021-9991. doi: 10.1016/J.JCP.2018.10.045.
- [20] Feilong Jiang, Min Xia, and Yaowu Hu. Physics-Informed Machine Learning for Accurate Prediction of Temperature and Melt Pool Dimension in Metal Additive Manufacturing. *3D Printing and Additive Manufacturing*, 11(4):e1679–e1689, August 2024. ISSN 2329-7662. doi: 10.1089/3dp.2022.0363. URL <https://www.liebertpub.com/doi/full/10.1089/3dp.2022.0363>. Publisher: Mary Ann Liebert, Inc., publishers.
- [21] Qiming Zhu, Zeliang Liu, and Jinhui Yan. Machine learning for metal additive manufacturing: predicting temperature and melt pool fluid dynamics using physics-informed neural networks. *Computational Mechanics*, 67(2):619–635, February 2021. ISSN 0178-7675, 1432-0924. doi: 10.1007/s00466-020-01952-9. URL <https://link.springer.com/10.1007/s00466-020-01952-9>.
- [22] Shilin Li, Gang Wang, Yuelan Di, Liping Wang, Haidou Wang, and Qingjun Zhou. A physics-informed neural network framework to predict 3D temperature field without labeled data in process of laser metal deposition. *Engineering Applications of Artificial Intelligence*, 120:105908, April 2023. ISSN 0952-1976. doi: 10.1016/j.engappai.2023.105908. URL <https://www.sciencedirect.com/science/article/pii/S0952197623000921>. 9 citations (Crossref) [2024-07-03] rate: 2.

- [23] Jibing Xie, Ze Chai, Luming Xu, Xukai Ren, Sheng Liu, and Xiaoqi Chen. 3D temperature field prediction in direct energy deposition of metals using physics informed neural network. *The International Journal of Advanced Manufacturing Technology*, 119(5):3449–3468, March 2022. ISSN 1433-3015. doi: 10.1007/s00170-021-08542-w. URL <https://doi.org/10.1007/s00170-021-08542-w>. 16 citations (Crossref) [2024-06-16].
- [24] Shuheng Liao, Tianju Xue, Jihoon Jeong, Samantha Webster, Kornel Ehmann, and Jian Cao. Hybrid thermal modeling of additive manufacturing processes using physics-informed neural networks for temperature prediction and parameter identification. *Computational Mechanics*, 72(3):499–512, September 2023. ISSN 0178-7675, 1432-0924. doi: 10.1007/s00466-022-02257-9. URL <https://link.springer.com/10.1007/s00466-022-02257-9>.
- [25] Atilim Gunes Baydin, Barak A. Pearlmutter, Alexey Andreyevich Radul, and Jeffrey Mark Siskind. Automatic differentiation in machine learning: a survey, 2015.
- [26] John Goldak, Aditya Chakravarti, and Malcolm Bibby. A new finite element model for welding heat sources. *Metallurgical Transactions B*, 15(2):299–305, June 1984. ISSN 0360-2141, 2379-0229. doi: 10.1007/BF02667333. URL <https://link.springer.com/10.1007/BF02667333>.
- [27] Sifan Wang, Shyam Sankaran, Hanwen Wang, and Paris Perdikaris. An Expert’s Guide to Training Physics-informed Neural Networks, August 2023. URL <http://arxiv.org/abs/2308.08468>. arXiv:2308.08468 [physics] keywords = Computer Science - Machine Learning, Mathematics - Numerical Analysis, Physics - Computational Physics, PINN,.
- [28] Salvatore Cuomo, Vincenzo Schiano Di Cola, Fabio Giampaolo, Gianluigi Rozza, Maziar Raissi, and Francesco Piccialli. Scientific Machine Learning Through Physics-Informed Neural Networks: Where we are and What’s Next. *Journal of Scientific Computing*, 92(3): 88, September 2022. ISSN 0885-7474, 1573-7691. doi: 10.1007/s10915-022-01939-z. URL <https://link.springer.com/10.1007/s10915-022-01939-z>.
- [29] Shashank Subramanian, Robert M. Kirby, Michael W. Mahoney, and Amir Gholami. Adaptive Self-supervision Algorithms for Physics-informed Neural Networks, July 2022. URL <http://arxiv.org/abs/2207.04084>. arXiv:2207.04084 [physics].
- [30] Sifan Wang, Shyam Sankaran, and Paris Perdikaris. Respecting causality is all you need for training physics-informed neural networks, March 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.
- [31] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An Imperative Style, High-Performance Deep Learning Library, December 2019. URL <http://arxiv.org/abs/1912.01703>. arXiv:1912.01703 [cs, stat].
- [32] Lu Lu, Raphaël Pestourie, Wenjie Yao, Zhicheng Wang, Francesc Verdugo, and Steven G. Johnson. Physics-Informed Neural Networks with Hard Constraints for Inverse Design. *SIAM Journal on Scientific Computing*, 43(6):B1105–B1132, January 2021. ISSN 1064-8275. doi: 10.1137/21M1397908. URL <https://epubs.siam.org/doi/10.1137/21M1397908>. Publisher: Society for Industrial and Applied Mathematics.
- [33] Haotian Chen, Christopher Batchelor-McAuley, Enno Kätelhön, Joseph Elliott, and Richard G. Compton. A Critical Evaluation of Using Physics-Informed Neural Networks for Simulating Voltammetry: Strengths, Weaknesses and Best Practices. *Journal of Electroanalytical Chemistry*, 925:116918, November 2022. ISSN 1572-6657. doi: 10.1016/j.jelechem.2022.116918. URL <https://www.sciencedirect.com/science/article/pii/S1572665722009109>. 10 citations (Crossref) [2024-06-02].

Appendix A Governing Equations for 3D implementation

In the 3D implementation, there are some slight changes to the governing equations as detailed in this section. The key difference is the additional dimension in the thickness directions. Figure A1 indicates the axes used in the 2D and 3D implementations, respectively.

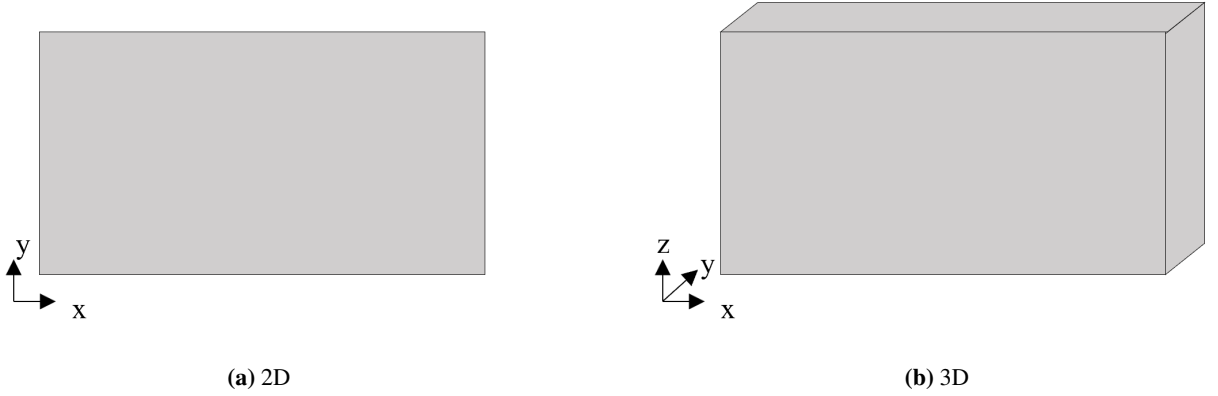


Figure A1: Axes in the 2D and 3D implementations

Equations (2) to (4) for 3D are rewritten as:

$$T \frac{\partial T}{\partial t} \frac{\partial \rho C_p}{\partial T} + \rho(T) C_p(T) \frac{\partial T}{\partial t} - \frac{\partial k}{\partial T} \left(\left(\frac{\partial T}{\partial x} \right)^2 + \left(\frac{\partial T}{\partial y} \right)^2 + \left(\frac{\partial T}{\partial z} \right)^2 \right) - k(T) \left(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} + \frac{\partial^2 T}{\partial z^2} \right) = \dot{e}_{\text{gen}} \quad (\text{A1})$$

$$\rho C_p \frac{\partial T}{\partial t} - k \left(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} + \frac{\partial^2 T}{\partial z^2} \right) = \dot{e}_{\text{gen}} \quad (\text{A2})$$

$$\dot{e}_{\text{gen}} = \frac{6\sqrt{3}\eta P}{\pi abc} e^{-3 \left(\frac{(x-x_{\text{laser}})^2}{a^2} + \frac{(y-y_{\text{laser}})^2}{b^2} + \frac{(z-z_{\text{laser}})^2}{c^2} \right)} \quad (\text{A3})$$

The expression for the characteristic dimensions is expanded as following for the additional width dimension:

$$\begin{cases} x &= l_c x^* \\ y &= l_c y^* \\ z &= l_c z^* \\ t &= t_c t^* \\ T &= T_c T^* \end{cases} \quad (\text{A4})$$

Subsequently, the terms in Equations (12) to (14) are modified to the following:

$$\begin{aligned} \tilde{F}(\hat{T}^*, \mathbf{x}^*, t^*) &= \frac{\partial \hat{T}^*}{\partial t^*} - \frac{k}{\rho C_p} \frac{t_c}{l_c^2} \left(\frac{\partial^2 \hat{T}^*}{\partial x^{*2}} + \frac{\partial^2 \hat{T}^*}{\partial y^{*2}} + \frac{\partial^2 \hat{T}^*}{\partial z^{*2}} \right) \\ &\quad - \frac{1}{\rho C_p} \frac{t_c}{T_c} \frac{6\sqrt{3}\eta P}{\pi abc} e^{-3 \left(\frac{(x^* - (x_0^* + v^* t^*))^2}{a^2} + \frac{y^{*2} l_c^2}{b^2} + \frac{z^{*2} l_c^2}{c^2} \right)} \end{aligned} \quad (\text{A5})$$

$$\tilde{B}(\hat{T}^*, \mathbf{x}^*, t^*) = \begin{cases} \pm \frac{\partial T^*}{\partial n^*} - \frac{h}{k} l_c (\hat{T}^* - T_{\text{amb}}^*) - \frac{\sigma \epsilon}{k} T_c^3 l_c (\hat{T}^{*4} - T_{\text{amb}}^{*4}) \\ \frac{\partial T^*}{\partial y^*} - \frac{h_{\text{force}}}{k} l_c (\hat{T}^* - T_{\text{amb}}^*) \end{cases} \quad (\text{A6})$$

$$\tilde{I}(\hat{T}^*, \mathbf{x}^*, 0) = \hat{T}_{t=t_0}^* - T_{\text{ref}}^* \quad (\text{A7})$$

The laser on-off switch is updated following the new expression in Equation A3

$$\dot{e}_{\text{gen}}^* = \text{laser_switch}(t^*) \frac{1}{\rho C_p} \frac{t_c}{T_c} \frac{6\eta\sqrt{3}P}{\pi abc} e^{-3 \left(\frac{(x^* - (x_0^* + v^* t^*))^2 l_c^2}{a^2} + \frac{(y^* - y_{\text{laser}}^*)^2 l_c^2}{b^2} + \frac{(z^* - z_{\text{laser}}^*)^2 l_c^2}{c^2} \right)} \quad (\text{A8})$$

The solid-void transition in 23 is updated to the following based on the change of axes system:

$$\begin{aligned} \tilde{\mathcal{F}} = & \tilde{\mathcal{F}}^{\text{solid}} \left(\left[\left(1 + e^{\alpha_z \Delta x_b^*} \right) \left(1 + e^{-\alpha_z \Delta z_b^*} \right) \right]^{-1} + \left[1 + e^{\alpha_z \Delta z_b^*} \right]^{-1} \right) \\ & + \tilde{\mathcal{F}}^{\text{air}} \left[\left(1 + e^{-\alpha_z \Delta x_b^*} \right) \left(1 + e^{-\alpha_z \Delta z_b^*} \right) \right]^{-1} \end{aligned} \quad (\text{A9})$$

where

$$\begin{cases} \Delta x_b^* &= x^* - x_{\text{boundary}}^* \\ \Delta z_b^* &= z^* - z_{\text{boundary}}^* \end{cases}$$

Appendix B Configurations and Setup

The material properties and process settings and hyperparameters (bespoke and standard) are listed in Table B1, B2, and B3, respectively. All training and simulations are run on a local workstation that is equipped with an Intel i7-12700k CPU and an Nvidia RTX4070 (12GB VRAM) GPU.

Table B1: List of parameters used in training and simulation

	Parameter	Nomenclature	Value	Unit
Material (solid)	Heat conductivity	k	1.14E+01	$mW/(mm\ K)$
	Density	ρ	4.50E-09	$tonne/mm^3$
	Heat capacity	Cp	7.14E+08	$mJ/(tonne\ K)$
Material (air)	Heat conductivity	k_{air}	2.60E-02	$mW/(mm\ K)$
	Density	ρ_{air}	1.18E-12	$tonne/mm^3$
	Heat capacity	Cp_{air}	1.01E+09	$mJ/(tonne\ K)$
Process	Stefan-Boltzmann constant	σ	5.67E-11	$mW/(mm^2\ K^4)$
	Emissivity	ϵ	3.00E-01	—
	Convection coefficient (forced)	h_{forced}	4.50E-01	$mW/(mm^2\ K)$
	Convection coefficient	h	2.00E-02	$mW/(mm^2\ K)$
	Laser power	P	3.50E+05	mW
	Laser absorptivity	η	4.00E-01	—
	Laser speed	v_{laser}	1.00E+01	mm/s
	Laser spot diameter	ϕ_{laser}	2.00E+00	mm
	Goldak parameter (length)	a	1.00E+00	mm
	Goldak parameter (depth)	b	2.00E+00	mm
	Ambient temperature	T_{amb}	2.98E+02	K

Table B2: List of bespoke hyperparameters used in training and simulation

Hyperparameter	Nomenclature	Value	Unit
Characteristic length	l_c	ϕ_{laser}	mm
Characteristic time	t_c	ϕ_{laser}/v_{laser}	s
Characteristic temperature	T_c	3000	K
Laser switch rise-time parameter	α_t	200	—
Pointwise weight cut-off rate parameter	α_w	200	—
Pointwise weight cut-off offset parameter	β	0.05	—
Solid-void transition rate parameter	α_z	100	—
Causality stiffness	ϵ	1	—
Adaptive weight update rate	α	0.9	—

Table B3: List of hyperparameters used in training

Hyperparameter	Value
NN depth	3
NN width	128
Activation function	$\tanh()$
Optimisor	<i>Adam</i>
Learn rate	$2e-5$

Appendix C Convergence Study for ANSYS Mesh Resolution

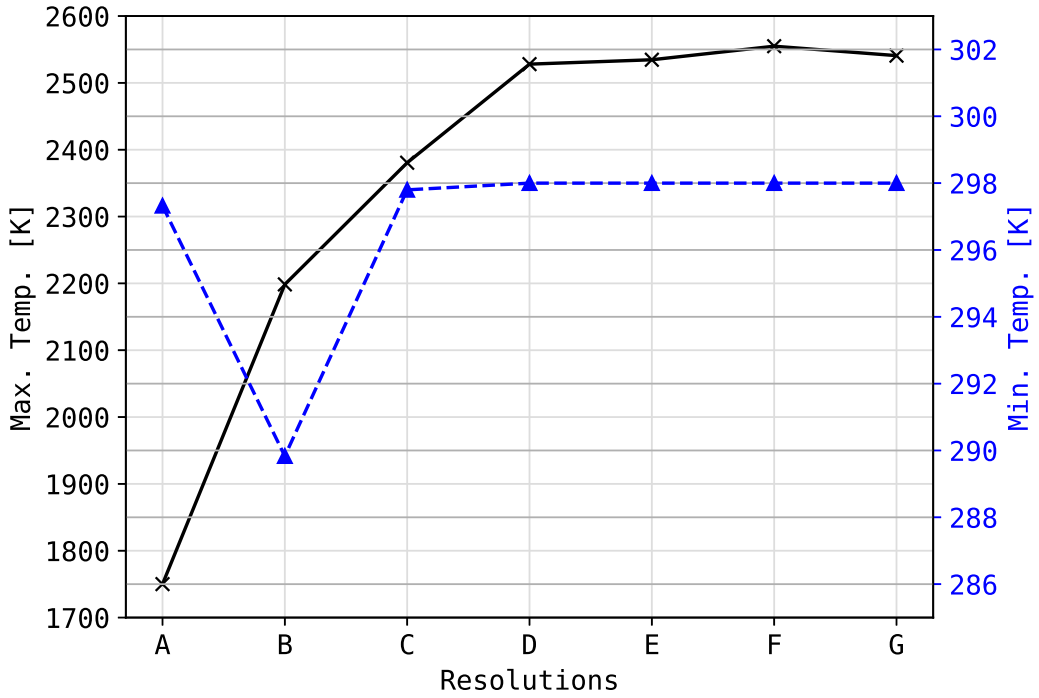


Figure C1: Mesh convergence study

Table C1: Configurations for mesh convergence study

Case	Mesh Resolution			Time Resolution [s]	Processing Time per Layer [s]
	Length [mm]	Height (Part) [mm]	Height (Subs.) [mm]		
A	0.3	0.5	2.5	0.015	68
B	0.15	0.25	2.5	7.5e-3	212
C	0.15	0.25	1.25	7.5e-3	228
D	0.15	0.05	0.625	7.5e-3	1619
E	0.15	0.05	0.625	3.75e-3	1231
F	7.5e-2	0.05	0.625	3.75e-3	2429