Physics-informed neural networks to accelerate heat transfer predictions in additive manufacturing

Junhyeok Kil*, Akshay J. Thomas**, Eduardo Barocio**, Ilias Bilionis**

*Department of Computer Science – Purdue University. West Lafayette, IN, USA.
**School of Mechanical Engineering – Purdue University. West Lafayette, IN, USA.

Abstract

Accurate and scalable prediction of thermal history is pivotal for optimizing print quality in extrusion deposition additive manufacturing (EDAM). This paper investigates the applicability of physics-informed neural networks (PINNs) for 3D heat transfer analysis during the additive manufacturing process. As finite element method (FEM) requires increasingly fine meshes and smaller time increments to achieve higher accuracy, its computational cost rises substantially. PINNs offer a more scalable solution by eliminating the need for meshes and time increment schemes. We achieve this by recasting the solution to the differential equation as a stochastic minimization problem. While the current focus is a single forward problem, the paper sets the groundwork for future research into parametric problems, where PINNs demonstrate their full potential by efficiently solving a range of scenarios under varying initial, boundary conditions, and material properties. Our results show that PINNs can be used to solve the 3D heat transfer equation on evolving geometries without the need for spatial discretization and time-stepping schemes. This study showcases the growing relevance of PINNs in manufacturing simulations, with future implications for real-time control and optimization in advanced manufacturing.

1. Introduction

Extrusion deposition additive manufacturing (EDAM) with fiber-reinforced polymers is a popular technology for producing large-scale parts in sectors like aerospace and automobile [1]. However, achieving optimal thermomechanical performance in EDAM parts remains challenging due to the complexity in accurately predicting temperature evolution during printing process. This is an important problem because the temperature distribution significantly impacts the material properties, which in turn affects the final part's structural performance [2].

The finite element method (FEM) is a numerical approach for predicting heat transfer in EDAM [3]. FEM discretizes the geometry with meshes and uses time increment schemes to compute transient conduction, convection, and radiation effects. FEM delivers reliable solutions but faces high computational costs when engineers refine meshes or reduce time increments to improve accuracy [4]. This becomes further problematic for fiber-reinforced composites, where anisotropic conduction adds complexity to the simulation [2].

The rise of graphical processing units (GPUs) has spurred the development of physics-informed neural networks (PINNs) as an alternative solution [5]. PINNs embed physical laws like the heat equation written as partial differential equations (PDEs) directly into a single neural network-based loss function. PINNs do not require explicit meshing or time increment schemes. Instead, it relies on automatic differentiation and flexible sampling in space and time, offering a novel approach to solving physical problems, even without observational data [6]. Raissi et al. introduced PINNs for solving forward and inverse problems governed by PDEs [5]. Since then, numerous studies have demonstrated PINNs' efficacy in simulating complex physical systems, showing promise across multiple domains such as fluid dynamics, solid mechanics, electromagnetics, and additive manufacturing [7, 8, 9, 10].

SAMPE Conference Proceedings. Indianapolis, IN, May 19-22, 2025. Society for the Advancement of Material and Process Engineering – North America.

DOI: https://doi.org/10.33599/nasampe/s.25.0137

A recent work in PINNs shows potential to handle evolving geometries and boundary conditions, which are common in EDAM [4]. They proposed a novel loss function incorporating a nested summation framework, enabling PINNs to respect causality while effectively tracking evolving domains. Our study builds on top of this by extending the application of PINNs from 2D to 3D domains. The focus is on modeling a single bead deposition with evolving geometries and complex boundary conditions in EDAM. By handling evolving geometries and boundary conditions within a 3D context, this research evaluates the accuracy, scalability, and computational efficiency of PINNs for realistic manufacturing scenarios.

2. Problem Formulation

We model the heat transfer in EDAM as a transient heat conduction problem in a 3D domain. The transient heat transfer during the EDAM process is governed by the heat conduction equation:

$$\rho C \frac{\partial u}{\partial t} = \nabla \cdot (k \nabla u) + Q$$

where $u(x, y, z, t)$ is the temperature at position (x, y, z) and time t. The variable ρ is the material density, C is the specific heat capacity, k represents the thermal conductivity tensor, and Q denotes the internal heat generation rate. In fiber-reinforced polymers, the thermal conductivity tensor is anisotropic, with $k = diag(k_{xx},\ k_{yy},\ k_{zz})$, where k_{xx}, k_{yy}, and k_{zz} are the thermal conductivities in the principal directions of the material. For this study, we assume $Q = 0$, i.e., there is no internal heat generation.

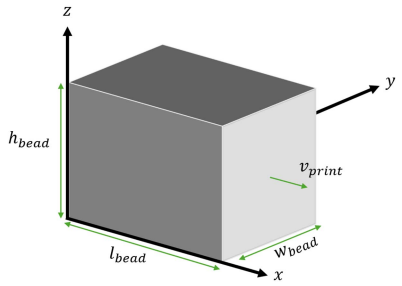

Figure 1: Domain of the Bead in Study

The domain has dimensions x_{max}, y_{max}, and z_{max}, representing the length, width, and height of the bead, respectively as shown in Figure 1. The deposition process is characterized by a moving heat source that advances along the x-axis at a constant print speed v_{print}, depositing material with width w_{bead} and height h_{bead}. This motion introduces an evolving boundary condition along the deposition front, which progresses with time. The single-layer setup is representative of the initial deposition process, where the thermal history is influenced primarily by the interaction of the extruded material with the ambient environment and the print bed. Multi-layer effects, such as residual heat retention from previously deposited layers, are not considered in this work. We are currently working on this aspect.

To compare the performance of PINNs, we analyze 3 distinct cases that progressively increase the complexity of the boundary conditions and geometry. Each case represents realistic scenarios in EDAM and evaluates the ability of PINNs to handle evolving geometries and mixed boundary conditions. In all cases, the initial temperature distribution within the domain, $u(x, y, z, t = 0)$, is assumed to be uniform and equal to the ambient temperature since there is no multi-layer deposition.

Case 1: Evolving Geometry with Evolving Dirichlet Boundary Condition

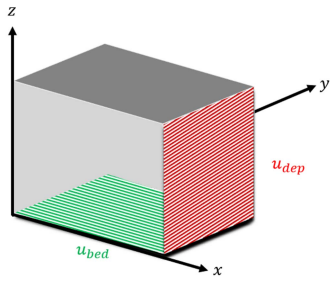

Figure 2: Illustration of Case 1

In Case 1, the geometry evolves dynamically as the deposition front progresses along the x-axis. The deposition front is governed by a Dirichlet boundary condition where the temperature is fixed at the deposition temperature, $u = u_{dep}$. The print bed is maintained at a fixed bed temperature, $u = u_{bed}$, as shown in Figure 2. All other surfaces are subject to adiabatic Neumann boundary conditions ($\frac{\partial u}{\partial n} = 0$), representing insulated boundaries, with n denoting the direction normal to the boundary. This case isolates the influence of evolving geometry and fixed Dirichlet boundary conditions on the thermal field.

Case 2: Geometry with Evolving Dirichlet and Neumann Boundary Conditions

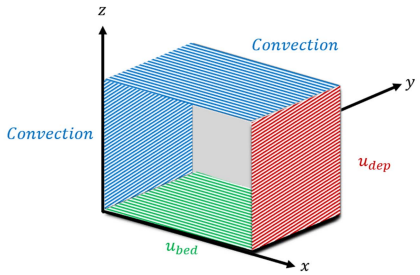

Figure 3: Illustration of Case 2

Case 2 extends the complexity of Case 1 in the previous example, by introducing Neumann boundary conditions on additional surfaces to model heat loss through convection. Just like in Case 1, the deposition front retains the Dirichlet boundary condition, where the temperature is fixed at the deposition temperature, $u = u_{dep}$. Similarly, the print bed remains at the fixed bed temperature, $u = u_{bed}$. However, the back surface (opposite the deposition front) and the top surface are now subject to Neumann boundary conditions to account for convective heat loss. These conditions are expressed as $k\frac{\partial u}{\partial n} = -h(u - u_{\infty})$, where h is the heat transfer coefficient and u_{∞} is the ambient temperature. The lateral surfaces continue to have adiabatic Neuman boundary conditions ($\frac{\partial u}{\partial n} = 0$). The addition of convection on the back and top surface introduces new heat dissipation mechanisms, with competing boundary conditions that reflect the interplay between fixed Dirichlet and evolving Neumann boundary conditions. From Figure 3, these competing boundary conditions occur at the intersection of the blue and green surfaces, and at the intersection of the red and blue surfaces.

Case 3: Evolving Geometry with Dirichlet Boundary Condition and Neumann Boundary Conditions Elsewhere

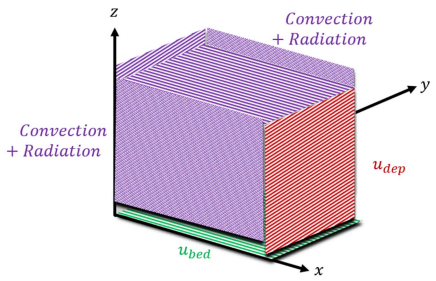

Figure 4: Illustration of Case 3

Case 3 further generalizes the boundary conditions by applying Neumann boundary conditions representing both convection and radiation on all surfaces except the deposition front and the bed surface as shown in Figure 4. The addition of radiation effects increases the complexity of the thermal analysis. Radiation heat loss is expressed as $k\frac{\partial u}{\partial n} = -h(u - u_\infty) - \epsilon\sigma(u^4 - u_\infty^4)$, where ϵ is the emissivity of the surface and $\sigma = 5.67 \times 10^{-8} W/m^2K^4$ is the Stefan-Boltzmann constant. By combining convection and radiation on all surfaces, this case best approximates real-world conditions.

PINNs are often plagued with difficulty in training. To alleviate the training issue, we follow the recommendations noted in [9]. The most important step is to non-dimensionalize the governing equations achieved by scaling the fundamental quantities to dimensionless forms. Time is scaled as $\tilde{t} = \frac{t}{t_{max}}$, spatial coordinates are scaled as $(\tilde{x}, \tilde{y}, \tilde{z}) = \left(\frac{x}{x_{max}}, \frac{y}{y_{max}}, \frac{z}{z_{max}}\right)$, and the temperature is scaled relative to the maximum deposition temperature as $\tilde{u} = \frac{u}{u_{dep}}$. As a consequence of these transformations, thermal diffusivity is similarly scaled, yielding dimensionless parameters $\alpha_{xx}, \alpha_{yy}, \alpha_{zz}$ for thermal conductivity, defined as $\alpha_{ii} = k_{ii}/(\rho C x_{max}^2/t_{max})$.

3. Methodology

3. 1 Loss Function and Estimator

In this work, we consider a total loss function $\mathcal{L}$ [4] expressed as:

$$\mathcal{L} = \int_0^T \ell(t, x(t); \theta) dt,$$

where $\ell(t, x(t); \theta)$ is the loss at a given time t, which depends on the evolving spatial domain $x(t)$. To make this equation amenable to optimization algorithms, we seek an unbiased estimator of the integral. An unbiased estimator of this integral is the sample average over time. An unbiased estimator is essential for convergence. Sampling M time points in the interval $[0, T]$ and letting $|T| = T$ denote the total time, we obtain:

$$\mathcal{L}_E = \frac{|T|}{M} \sum_{m=1}^{M} \ell(t_m, x(t_m); \theta).$$

The term $\ell(t_m, x(t_m); \theta)$ is chosen as the squared residual of the PDE and boundary conditions at the sampled times. Discretizing the temporal integral into N_t time steps and spatial losses into N_x collocation points per time step, the loss function becomes:

$$\mathcal{L}_E = \frac{1}{N_t}\sum_{m=1}^{N_t}\left(\frac{\lambda_{PDE}}{N_{PDE}}\sum_{i=1}^{N_{PDE}} Res^2(t_m, x_i(t_m)) + \frac{\lambda_D}{N_D}\sum_{j=1}^{N_D}\left[u_{pred}\left(t_m, x_j(t_m)\right) - u_{specified}\left(t_m, x_j(t_m)\right)\right]^2 + \frac{\lambda_N}{N_N}\sum_{k=1}^{N_N}\left[n \cdot k\nabla u_{pred}\left(t_m, x_k(t_m)\right) - q_{specified}\left(t_m, x_k(t_m)\right)\right]^2\right),$$

where each term corresponds to contributions from the PDE residual, Dirichlet boundary conditions, and Neumann boundary conditions. The coefficients λ_{PDE}, λ_D, and λ_N weight the contributions of each term to the total loss. Similarly, convection and radiation losses are added as:

$$\mathcal{L}_{Conv} = \frac{\lambda_{Conv}}{N_{Conv}}\sum_{m=1}^{N_t}\sum_{i=1}^{N_{Conv}}\left[k\frac{\partial u_{pred}}{\partial n}\left(t_m, x_i(t_m)\right) + h\left(u_{pred}\left(t_m, x_i(t_m)\right) - u_\infty\right)\right]^2$$

and

$$\mathcal{L}_{Rad} = \frac{\lambda_{Rad}}{N_{Rad}}\sum_{m=1}^{N_t}\sum_{i=1}^{N_{Rad}}\left[k\frac{\partial u_{pred}}{\partial n}\left(t_m, x_i(t_m)\right) + \epsilon\sigma\left(u_{pred}^4\left(t_m, x_i(t_m)\right) - u_\infty^4\right)\right]^2.$$

3.2 Governing PDE and Residual

The neural network outputs the temperature $u\,(x, y, z, t)$, and to satisfy the governing PDE, we explicitly compute derivatives of u_{pred} with respect to spatial and temporal variables using automatic differentiation. These derivatives are used to construct the residual of the heat equation, which we denote as $Res(x, y, z, t)$, given by:

$$Res(x, y, z, t) = \rho C\frac{\partial u_{pred}}{\partial t} - \nabla \cdot \left(k\nabla u_{pred}\right)$$

The residual is incorporated into the loss function defined above to ensure that the predicted solution satisfies the governing equations at collocation points sampled throughout the domain.

The boundary and initial conditions are enforced by additional terms in the loss function. For Dirichlet boundary conditions, the network directly minimizes the difference between the predicted and specified boundary temperatures: $u_{pred}\left(t, x(t)\right) - u_{specified}\left(t, x(t)\right)$. For Neumann boundary conditions, the derivatives of the predicted temperature normal to the boundary are evaluated: $n \cdot k\nabla u_{pred}\left(t, x_k(t)\right) - q_{specified}\left(t, x_k(t)\right)$. Convection and radiation effects are incorporated as specific Neumann boundary conditions.

3.3 Sampling

PINNs use automatic differentiation to compute derivatives of the solution with respect to space and time, which allows flexible sampling in the domain. Flexible sampling refers to the ability to select points dynamically from spatial and temporal domain during training, without being constrained by a fixed mesh or grid structure, as is typical in FEM. This feature allows PINNs to adaptively sample regions of interest within the domain, such as areas with sharp gradients or rapidly evolving features. These sampling points are called collocation points

Collocation points are sampled dynamically during training using samplers that distribute points across the domain and on the boundaries. For example, one class of samplers generates points in the entire 3D volume, ensuring that the PDE residual is evaluated at various spatial and temporal locations (Figure 5).

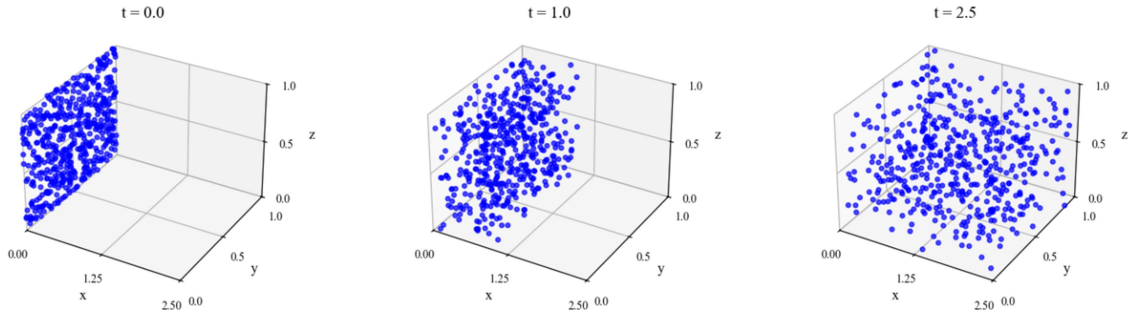

Figure 5: Illustration of domain collocation points sampled at different time steps

Similarly, boundary-specific samplers focus on sampling points on specific boundaries to enforce various boundary conditions (Figure 6, Figure 7).

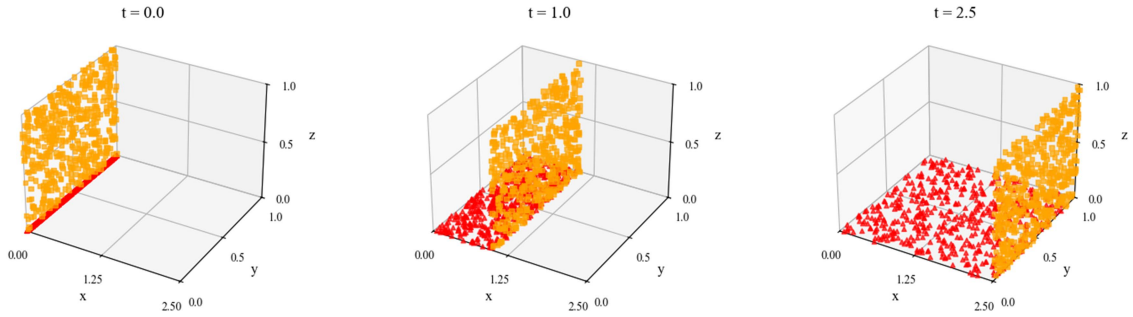

Figure 6: Illustration of collocation points for bed temperature (red) and deposition front (orange) sampled at different time steps

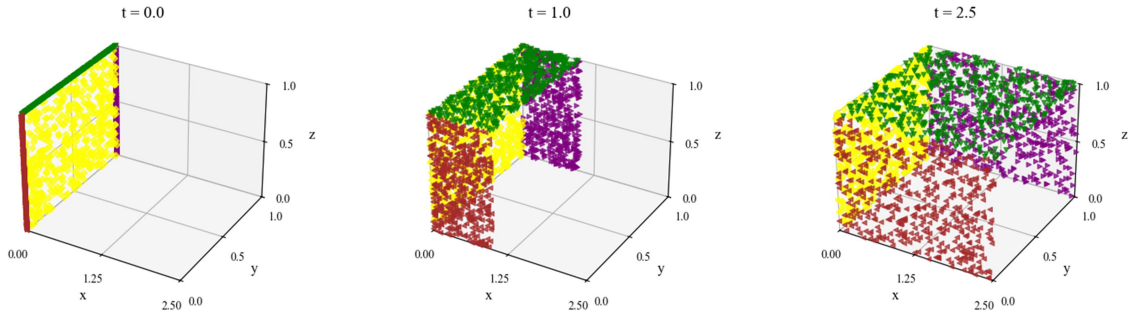

Figure 7: Illustration of collocation points for multiple surfaces of the geometry sampled at different time steps

The training process follows a novel approach where time is sampled first, followed by space [4]. For each sampled time step t_k, the collocation points are generated within the spatial domain defined at t_k. This sequential collocation sampling method naturally biases the learning of the solution at earlier times. This is facilitated by the fact that the variance of the loss estimator increases as the geometry evolves. The variance of the estimator increases as the geometry evolves since the collocation point density decreases (number of collocation points remains the same but the volume increases). This implies that the stochastic optimizer will converge for smaller times before converging for larger time, thereby enforcing causality.

3.4 Neural Network Architecture and Training

The neural network used in this study is a fully connected feedforward network with 4 input nodes corresponding to the time and spatial coordinates (t, x, y, z). The architecture consists of 4 hidden layers, each with 238 neurons, and uses the hyperbolic tangent $(tanh)$ activation function. The network outputs the temperature $u(t, x, y, z)$ as a scalar.

The training process uses the Adam [11] optimizer with a learning rate of 5×10^{-4}. PINNs are trained for 200,000 iterations with a batch size of 512 for spatial sampling and 8 for temporal sampling. The selection of number of iterations and batch sizes was guided by empirical observations from prior studies [4, 5]. We believe these settings balance computational efficiency with stable convergence. The network weights are initialized using a reparameterization strategy [12] to improve convergence. Fourier embeddings are used to enhance the representation of high-frequency components in the solution [4, 13]. The sampling and neural network training workflow is summarized in Figure 8.

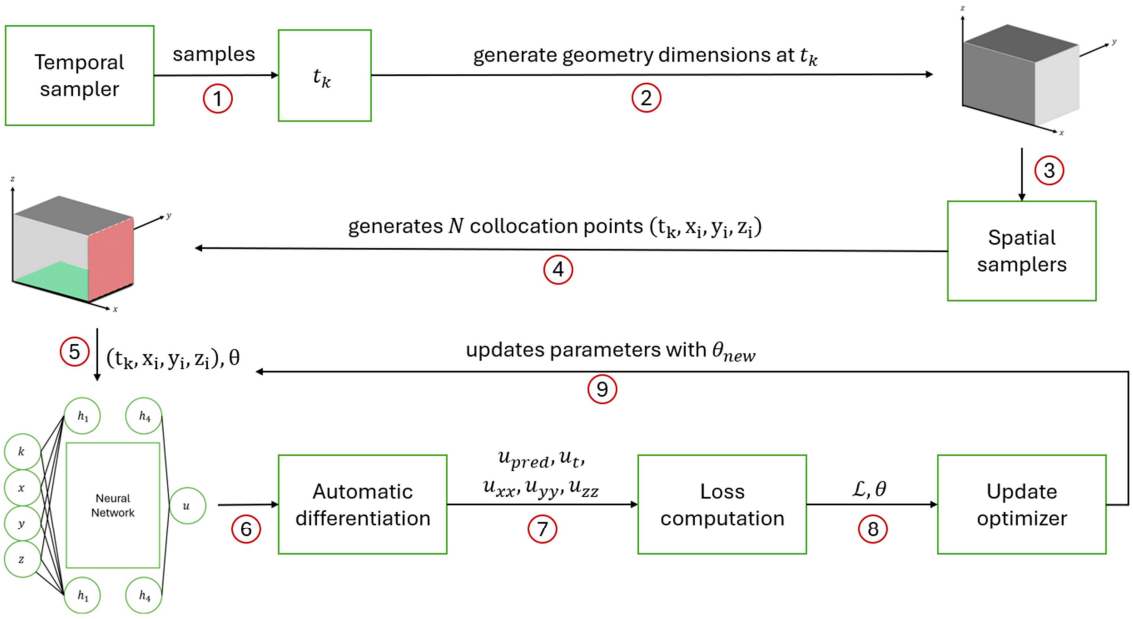

Figure 8: Summary of the methodology

3.5 Computational Times and Hardware Configuration

The computational times for the training and evaluation of the PINNs solver across the 3 cases were measured. Training times were consistent across cases, with Case 1 taking 27 minutes and 23 seconds, Case 2 taking 27 minutes and 27 seconds, and Case 3 taking 27 minutes and 14 seconds. The evaluation phase for all cases was under 1 minute. The simulations were performed on a Windows Subsystem for Linux (WSL) Ubuntu environment using hardware that included a 13th Gen Intel® Core™ i7-13620H processor operating at 2.4 GHz, 32.0 GB of RAM, and an NVIDIA GeForce RTX 4060 Laptop GPU with 8GB of memory. The CUDA version used was 12.6, and the driver version was 560.70.

4. Results

We evaluate the temperature profiles of the PINNs predictions in this work for the 3 cases. We look at temperature profiles predicted by the PINNs solver and investigate if it follows the physics of the problem. While we do not compare the solutions of our solver to FEM solutions, we believe we are on the right path to obtaining the right solutions. We recreate the same examples from [4] and observe that qualitatively we obtain very similar solutions. The time-weighted average loss terms for all cases are

shown in Figure 10. Notice that across all 3 cases, u_{dep} and u_{bed} exhibit loss values that have sufficiently leveled off, indicating convergence. Their final loss values 5.44×10^{-5} for u_{bed} and 8.10×10^{-3} for u_{dep}, are small enough to suggest that further training yield negligible gains. Thus, training was halted at 200,000 iterations as further improvements became marginal.

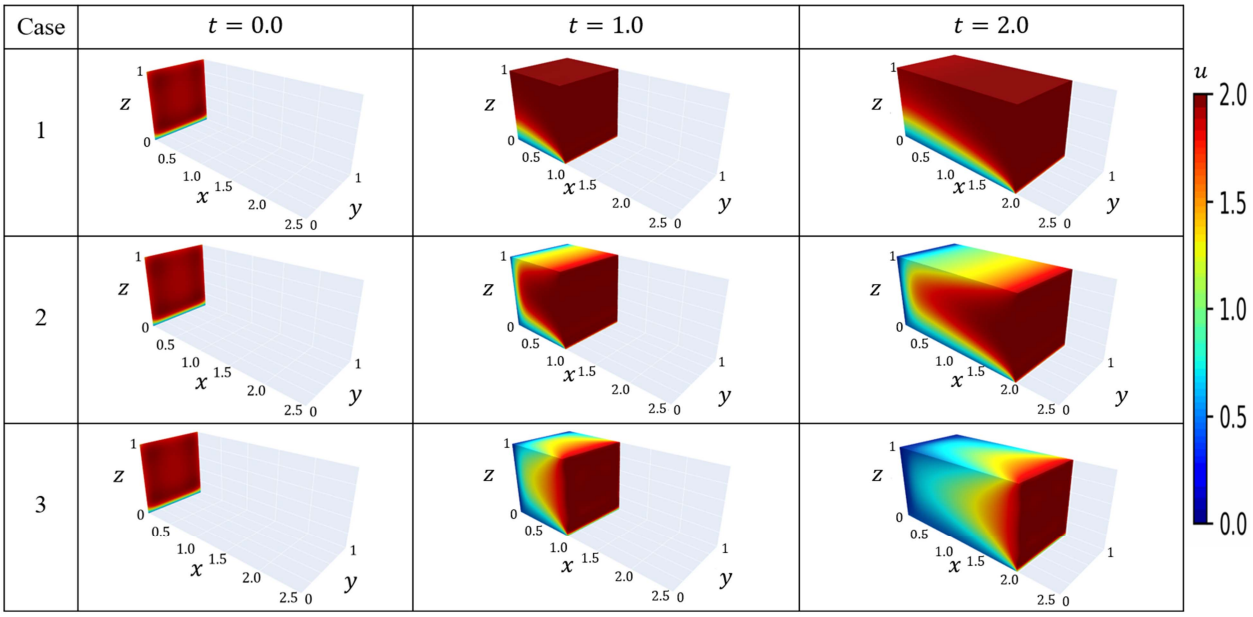

Figure 9: Visual summary of the results

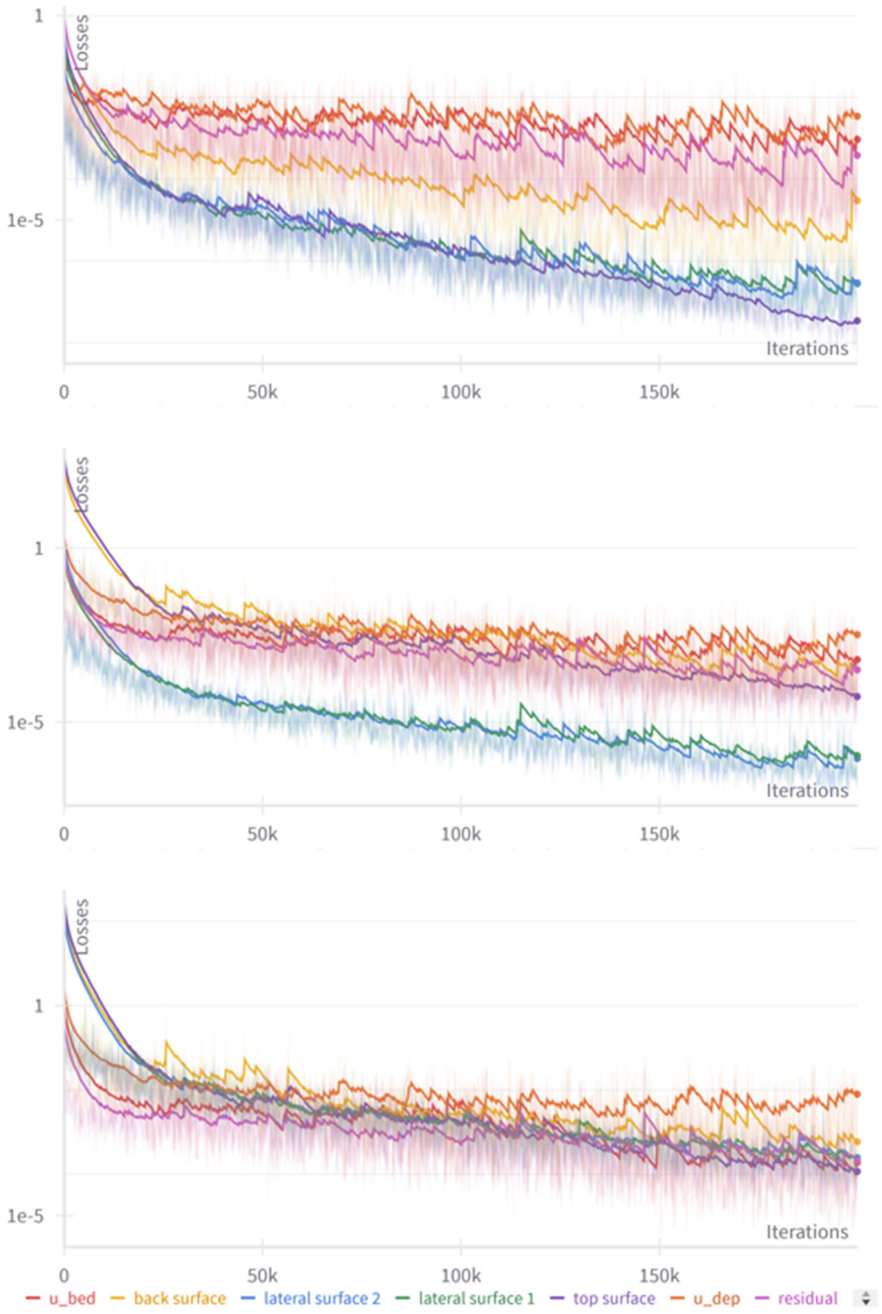

Figure 10: Time-weighted average loss terms for Case 1, 2 and 3 respectively. The darker lines are the average values.

Case 1: Evolving Geometry with Evolving Dirichlet Boundary Condition

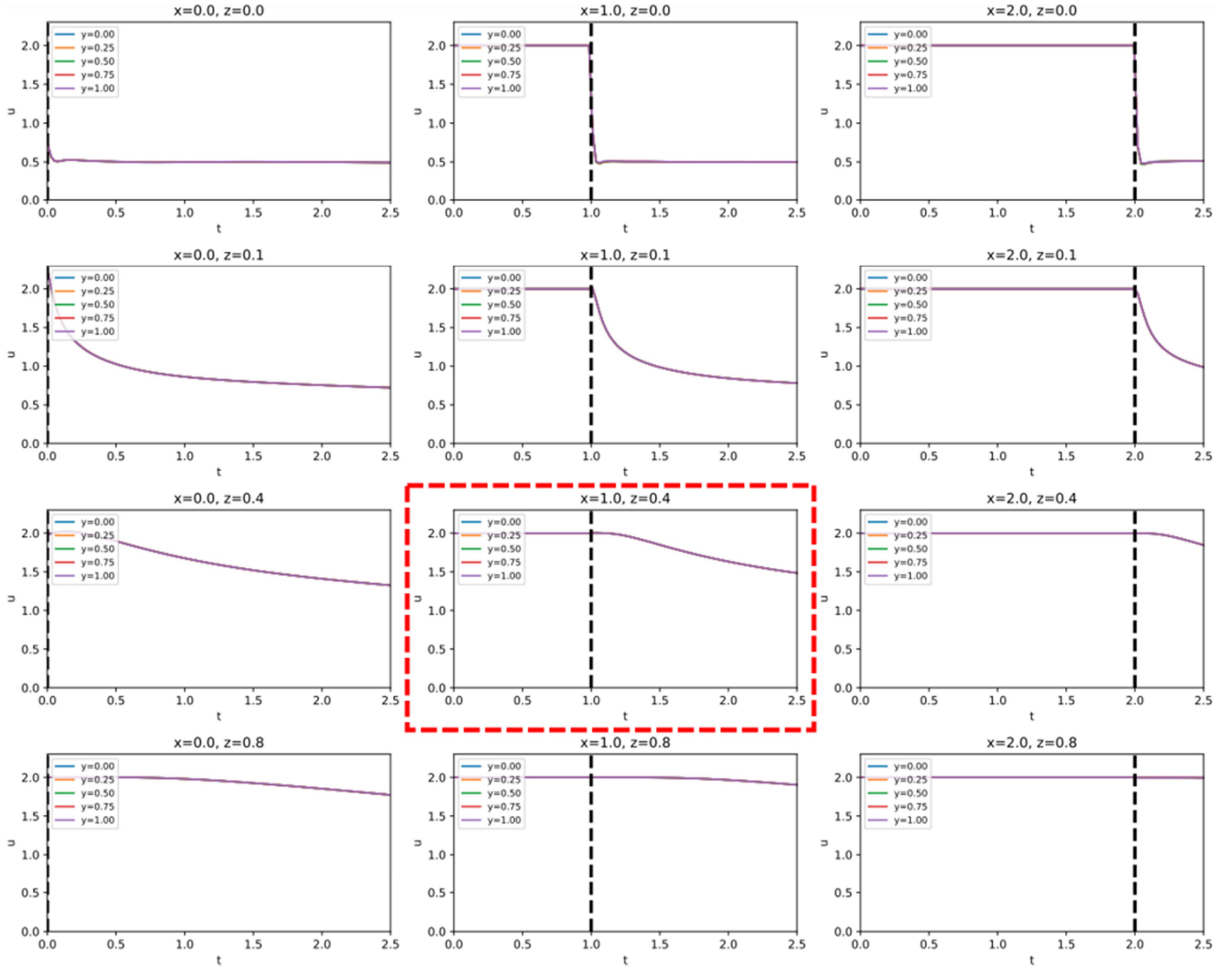

Figure 11: Case 1 Heat predictions at various locations at different slices of y. The vertical dotted line represents the time the particular location has been activated. The field is set to the maximum value before activation.

In the 3D solution figure for Case 1 (Figure 9, row 1), the domain before the material is deposited is visually "empty", representing the absence of thermal activity in regions that have not been activated. This is consistent with the physical process where the material deposition progresses along the x-axis, leaving regions behind it influenced by thermal processes while the regions ahead remain inactive. Correspondingly, in the temperature vs. time plot (Figure 11), the temperature is intentionally fixed to u_{dep} before the activation time.

To help understand Figure 11, we use the subplot at row 3 column 2 as an example. This subplot represents the evolution of heat predictions at the fixed locations of $x = 1.0$ and constant height at $z = 0.4$. This location corresponds to the red square region indicated in Figure 12.

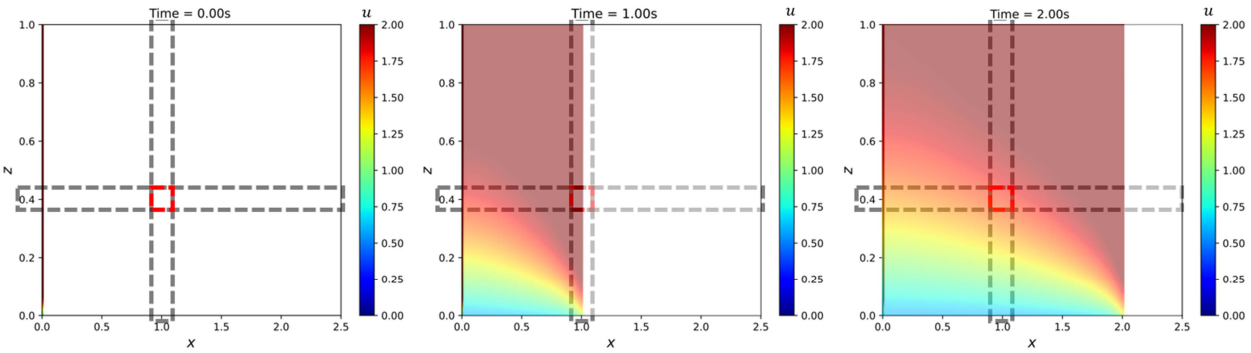

Figure 12: Supplementary visual aid for understanding heat prediction subplots

Upon material deposition, points close to the Dirichlet boundary condition at the deposition front cool faster, as evident from the steeper gradients in the temperature vs. time plot. Conversely, points farther from the Dirichlet boundary condition retain their temperature for longer periods, displaying slower cooling rates.

Case 2: Geometry with Evolving Dirichlet and Neumann Boundary Conditions

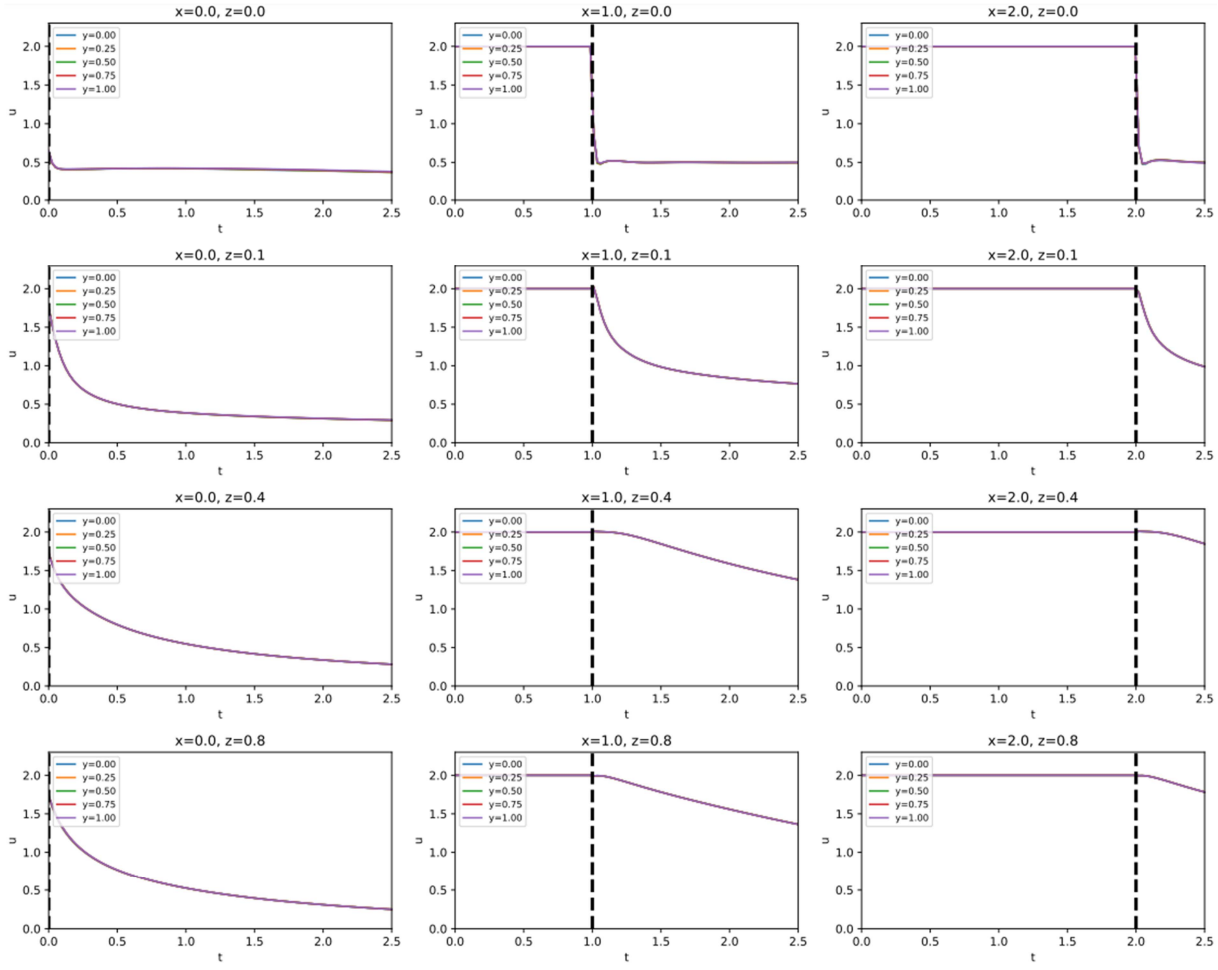

Figure 13: Case 2 Heat predictions at various locations at different slices of y. The vertical dotted line represents the time the particular location has been activated. The field is set to the maximum value before activation.

Case 2 builds upon Case 1 by introducing Neumann boundary conditions on the back and top surfaces to account for convective heat loss. Similar to Case 1, the volume before activation remains "empty" in the 3D solution figures (Figure 9, row 2), reflecting the absence of thermal activity before the deposition front reaches a particular region.

From Figure 13, we can see that after activation, points near the deposition front still cool rapidly, driven by the Dirichlet boundary condition, but the effect of convection becomes apparent on the back and top surfaces. These regions exhibit greater cooling rates compared to the insulated lateral surfaces. Points farther from the convective and deposition boundaries show slower cooling, maintaining higher temperatures over time. This difference highlights the interplay between the competing boundary conditions, as the convective heat loss accelerates cooling in specific regions while the insulated boundaries preserve heat.

Case 3: Evolving Geometry with Dirichlet Boundary Condition and Neumann Boundary Conditions Elsewhere

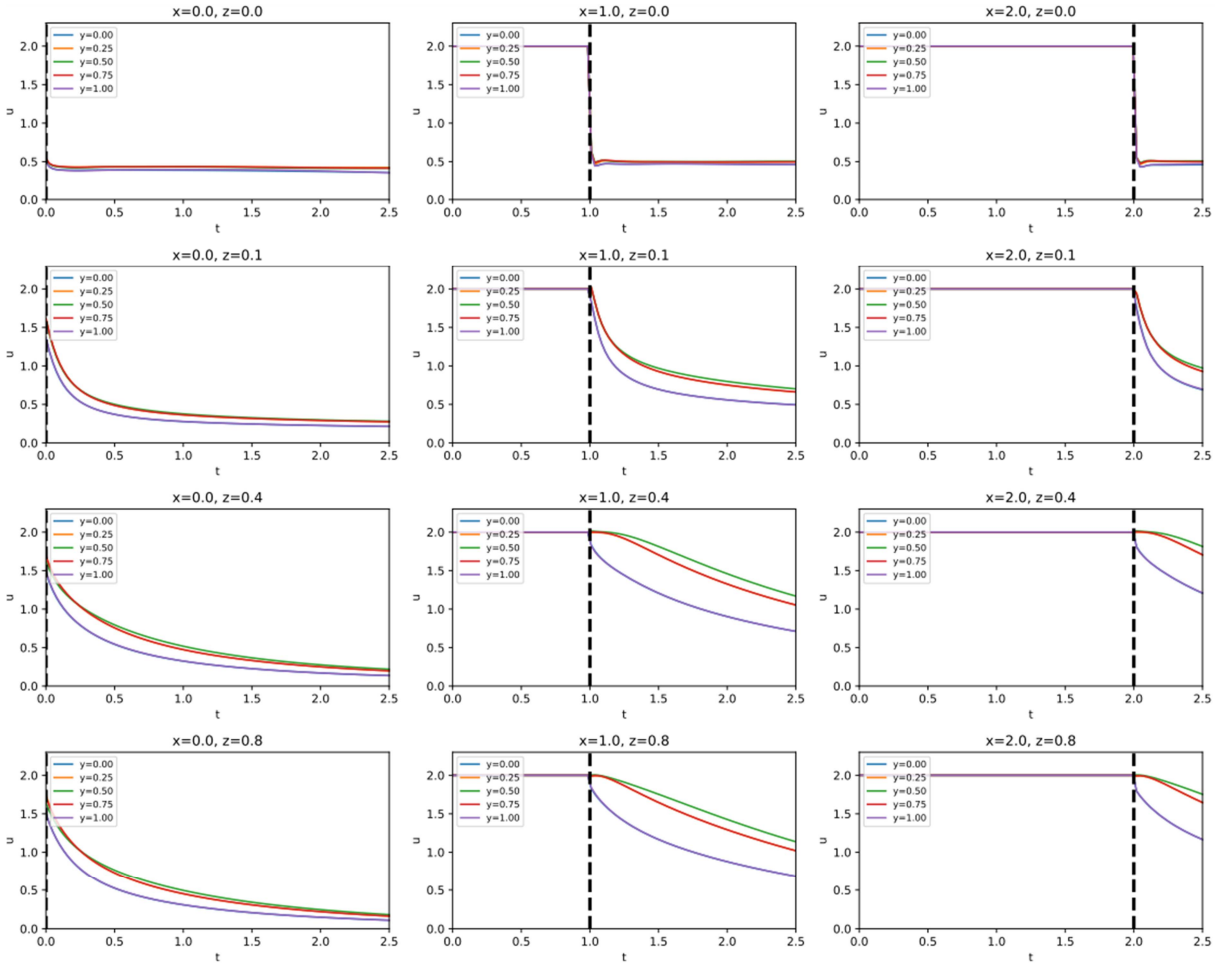

Figure 14: Case 3 Heat predictions at various locations at different slices of y. The vertical dotted line represents the time the particular location has been activated. The field is set to the maximum value before activation.

In Case 3, the thermal dynamics become even more intricate. Points near the deposition front continue to cool faster, driven by the Dirichlet boundary condition and the additional heat dissipations. The top and back surfaces exhibit accelerated cooling due to combined convection and radiation effects. The introduction of heat losses on all surfaces except the deposition front and print bed provides an

opportunity to better leverage the temperature vs. time plot (Figure 14). Each subplot of Figure 14 illustrates temperature evolution at different slices of y, allowing us to analyze the effects of lateral surface boundary conditions. Unlike Case 1 and Case 2 where the lateral surfaces are insulated with adiabatic Neumann boundary conditions, leading to uniform thermal behavior across all y-slices, the boundary conditions in Case 3 create distinct thermal profiles at different y-positions. This change introduces a temperature gradient across the y-direction, with points closer to the lateral surfaces cooling faster compared to those near the bead's core.

5. Conclusion

This study demonstrates PINNs as an alternative approach for solving heat transfer problems in EDAM. By reformulating the differential equation into a stochastic minimization problem, PINNs leverages the following advantages:

1. PINNs do not require spatial discretization or time increment schemes, reducing the complexities in meshing especially in complex or evolving geometries.
2. Nested sampling coupled with the evolving geometry naturally enforces causality and focuses learning the solutions for earlier times where the collocation point density is higher.
3. PINNs allow for the exact computation of the temperature gradients available via automatic differentiation.

To make PINNs available for parametric simulations, one must integrate the loss function introduced in this work over the parameters of interest. We believe this is the most exciting application for PINNs in manufacturing. The goal of this work was to evaluate the applicability of PINNs in 3D simulations. Our results give us some confidence that PINNs indeed is a viable solution for problems where the geometry evolves in 3D. Our immediate next step is to investigate the accuracy of our predictions by verifying them with a finite element solver. Further studies will focus on extending the methodology to parametric and more complex geometries. Additionally, investigating different sampling strategies and other network architectures may further enhance the accuracy and efficiency of PINNs solutions.

6. References

[1] Tekinalp, Halil L., Kunc, Vlastimil, Velez-Garcia, Gregorio M., Duty, Chad E., Love, Lonnie J., Naskar, Amit K., Blue, Craig A., & Ozcan, Soydan. *Highly oriented carbon fiber–polymer composites via additive manufacturing*. United States. https://doi.org/10.1016/j.compscitech.2014.10.009

[2] Brenken B, Barocio E, Favaloro A, Kunc V, Pipes RB. Development and validation of extrusion deposition additive manufacturing process simulations. Addit Manuf 2019;25:218–26.

[3] E. Barocio, P. Pibulchinda, A. J. Thomas, V. Kapre, and A. Franc, "Validated Simulation for Large Scale Additive Manufacturing.," CAMX 2022, 2022.

[4] A. J. Thomas, E. Barocio, I. Bilionis, R.B. Pipes, Toward parametric heat transfer solvers in additive manufacturing, Solid Freeform Fabrication 2024: Proceedings of the 35th Annual International Solid Freeform Fabrication Symposium – An Additive Manufacturing Conference.

[5] 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," J Comput Phys, vol. 378, pp. 686–707, 2019.

[6] J. Stiasny, S. Chevalier, S. Chatzivasileiadis, Learning without data: Physics-informed neural networks for fast time-domain simulation, in: 2021 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm), IEEE, pp. 634 438–443.

[7] H. Eivazi, M. Tahani, P. Schlatter, R. Vinuesa, Physics-informed neural networks for solving reynolds-averaged navier–stokes equations, Physics of Fluids 34 (2022).

[8] E. Haghighat, M. Raissi, A. Moure, H. Gomez, R. Juanes, A physics-informed deep learning framework for inversion and surrogate modeling in solid mechanics, Computer Methods in Applied Mechanics and Engineering 379 (2021) 113741.

[9] A. Khan and D. A. Lowther, "Physics Informed Neural Networks for Electromagnetic Analysis," in *IEEE Transactions on Magnetics*, vol. 58, no. 9, pp. 1-4, Sept. 2022, Art no. 7500404, doi: 10.1109/TMAG.2022.3161814

[10] S. Li, G. Wang, Y. Di, L. Wang, H. Wang, Q. 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 (2023) 105908.

[11] Kingma, D. P. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

[12] X. Glorot, Y. Bengio, Understanding the difficulty of training deep feedforward neural networks, in: Proceedings of the thirteenth international conference on artificial intelligence and statistics, JMLR Workshop and Conference Proceedings, pp. 249–256.

[13] S. Wang, H. Wang, P. Perdikaris, On the eigenvector bias of fourier feature networks: From regression to solving multi-scale pdes with physics-informed neural networks, Computer Methods in Applied Mechanics and Engineering 384 (2021) 113938.