Imperial College London
Department of Computing

Physics-informed Neural Networks in Additive Manufacturing

Christos Margadji

Pr. Andrew Davison
Dr. Paul Hooper

Submitted in partial fulfilment of the requirements for the
MSc Degree in Artificial Intelligence of Imperial College London.
September 2022

Abstract

Laser Powder Bed Fusion (LPBF) is revolutionising the automotive, space and other industries. However, it involves multi-physics processes that are non-deterministic in nature. This lack of stability hinders the quality of manufactured parts and substantially limits the technology. Computer vision and Machine Learning can provide in-situ monitoring for flaw detection, but real-time flaw localisation remains intractable. The industry is shifting towards digital twins (DT), that should enable further understanding of the system's dynamics for physics-informed decisions. One of the main pillars of DT are mechanistic models, frequently expressed by partial differential equations (PDE) and traditionally solved using finite methods. This forms a bottleneck: Given the fractal resolutions needed for multi-physics LPBF modelling, prohibitive solving times are required. In this report, a parameterised Physics-informed Neural Network (PINN) is trained without any data and in a self-supervised manner to approximate the heat diffusion PDE in the context of an LPBF setup. During inference it can instantly predict thermal profiles (3D structure) and steady-state melt pool characteristics (length, width, depth, area, volume) for any combination of laser focus, power and velocity. Other use cases are also discussed. The model takes into consideration the non-linear thermal-dependent properties of the assumed material, that being 316L stainless-steel. Its reliability is verified by comparing its outputs with theoretical ground-truth data.

Keywords: machine learning, additive manufacturing, physics-informed neural network, laser powder bed fusion, thermal modeling

Acknowledgements

This endeavour would not have been possible without Doctor Paul Hooper's expertise, insights and general advice. Our meetings and conversations truly helped me think of the problem from various perspectives.

A huge thanks goes to Professor Andrew Davison for accepting and unconditionally allowing me to pursue my ideas.

Next, I would like to acknowledge Sebastian Larsen for his guidance during the early stages of the project. In addition, I am very grateful to all other friends who have been around, either in-person or on-line, during my year at Imperial College London.

Last but not least, I express my warm gratitude to my family, especially my parents, for their tremendous support, constant encouragement and everything else.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Additive Manufacturing (AM) refers to the printing of parts on a layer-by-layer basis. It is gaining attention on a global scale, as it enables in-house rapid prototyping of complex geometries.[1] On the downside, the involved processes are non-deterministic, leading to inconsistencies in terms of geometric tolerance and part quality. To mitigate this, manufacturing engineers are needed to supervise the trial and error approach, to manually detect any defects that may occur. This significantly raises the cost of the technology and discourages its deployment for risk-averse applications. Researchers are focusing on in-situ monitoring methods to help develop flaw detection tools, in the hope of increasing the potential and reliability of the process.[2]

AM is particularly benefited from Artificial Intelligence (AI) due to its digital data-driven nature. Yang J. et al. published a survey on the current uses of AI in AM, identifying applications in all involved stages, from Design and Planning to Printing and Security.[3] Current drawbacks of AI in AM are discussed in [4]. In the context of part qualification, the most common obstacles are lack of computational power for low-latency automation and difficulties relating to data-sets. The scientific community tends to agree that totally optimising AM with conventional Machine Learning (ML) approaches becomes difficult when one considers the endless possibilities in process parameters. As such, generalisable

algorithms are needed that can be applied for any machine, material and part geometry.

Current trends revolve around self or semi-supervised solutions. Douglas B. et al. developed a network of extrusion 3D printers that collaboratively train a multi-head neural network by interchanging printing parameters and data obtained from cheap webcams.[5] Larsen S. et al. developed an anomaly detector based on a variational recurrent neural network (VRNN) that detects outliers in the temporal domain.[6] Pandiyan V. applied contrastive learning in the emissions from the process zone to extract meaningful semantics. Wand R. et al. developed a semi-supervised ML algorithm that can extract melt pool features from IR cameras and compared its predictions with measurements obtained from high-speed X-ray imaging.[7] However, pure data-driven ML methods fail to utilise prior knowledge that we may have in-hand concerning the processes involved.

Digital Twins (DT) in manufacturing aim to combine data knowledge with mechanistic models to enable multi-modal predictions that comply with the rules of Nature. Mechanistic models come in the form of partial differential equations (PDE); these are usually embedded into the system in the form of physics-based simulations, solved using finite methods. For instance, in [8] thermal predictions are fused with sensor data to detect process faults. Tang M. et al. used simpler geometry-based simulations to detect lack of fusion porosity, occurring when adjacent melt pools failed to overlap.[9] Wang W. et al. proposed a multi-physics modeling technique to predict keyhole pores given a process signature.[7] On the downside, the macroscopic and microscopic scale of the phenomena occurring during AM need high resolution meshes, making these methods suffer from prohibitive time and space complexities.

Rapid and computationally cheap simulation techniques are necessary for efficient usage of DT in feed-forward and in-line feedback control loops. Honarmandi P. et al. used Bayesian optimisation to fit the Eagar-Tsai model in the context of AM, achieving accurate depth estimation.[10] Ness K. et al. moved towards a more generic ML approach and trained an extremely randomised tree (ERT) using data generated from a simulation, in the aim of predicting temperature distributions and gradients.[11] Du Y. et al.

introduced a methodology that combined a genetic algorithm with mechanistic models and experimental data to reduce the occurrence of common defects.[12] Here, it should be noted that these methods heavily rely on data-sets which are expensive to acquire and label.

In this report, Physics-informed Neural Networks (PINN) are trained to instantly approximate solutions for the heat diffusion problem in the context of an LPBF setup. PINN are a class of deep neural networks (DNN) which utilise automatic differentiation to calculate and minimise the residual of any given PDE. This enables self-supervised training without ground-truth data, leading to mesh-free solutions. This highlights their suitability for AM, given the aforementioned difficulties with data.

It should be noted that literature exploring PINN in AM is relatively scarce. Earliest research comes from Zhu Q. et al. who used PINN to predict melt pool fluid dynamics.[13] Knuttel D. et al. explored the data dependency of 1D models and commented on the increased complexity of higher dimensional domains.[14] Shuheng L. et al. proposed a PINN for 3D thermal characterisation of AM using a 2D Gaussian surface heat flux source and constant material properties.[15] They also exhibited tactics to predict a range of unknown parameters of the process using the implemented framework. Finally, Hosseini E. et al. are currently working on developing a PINN solver for parametric 3D thermal analysis. To the authors knowledge this list is almost exhaustive and no published work analyses the surrogate model from an AI-engineering perspective.

1.1 Motivation and Objectives

The primary motivation of this research is to address the gap in literature concerning the AI-engineering practices for successful PINN implementation in the context of the heat diffusion PDE. To provide a qualitative and quantitative analysis, a PINN framework should be developed and tailored to the given problem and different approaches should be discussed. The ultimate goal is to eventually develop a real-time and robust depth-mapping algorithm which estimates the penetration depth of the melt pool at all locations of a layer scan. This could be used to indicate regions of interest based on their probability of having a defect in them; keyhole or lack of fusion porosity caused due to excessive or inadequate penetration. To approach the problem, a number of objectives have been set:

- Explore the available data and identify potential uses.
- Understand and formulate the heat diffusion problem in the context of AM and solve it using classic numerical methods.
- Implement a PINN to solve the formulated problem and investigate the effect of different design choices.
- Replicate a high-fidelity deposition experiment using a PINN.
- Parameterise a PINN to wrap multiple laser-dependent solutions in one surrogate model.
- Evaluate the accuracy of the developed PINN by comparing their outputs with ground truth data.

1.2 Statement of Originality

I certify that the intellectual content of this thesis is the product of my own work and that all the assistance received in preparing this thesis and sources have been acknowledged.

1.3 Structure of the Report

The rest of this report is organised as follows: Chapter 2 introduces the fundamental theories revolving around AI, PDE, AM and in particular LPBF. Chapter 3 details the methodology, the data engineering practices and the heat diffusion problem in the context of AM. Finite Difference Method (FDM) simulations are also used to tackle it. As an alternative, a PINN framework is then used and analysed in three different experiments. Chapter 4 discusses and compares the results of all experiments as a means of evaluation. A state-of-art method to extract meaningful information is also presented. Conclusions and suggested routes for future work are summarised in Chapter 5.

Chapter 2

Background Theory

2.1 Artificial Intelligence

In 1950, Turing A. published 'Computing Machinery and Intelligence' where he posed one of the most important questions of the 20th century; *Can machines think?*.[16] Since our civilisation started thinking about the problem we entered the era of AI, a term referring to the special branch of computer science that attempts to mimic human intelligence and decision making. More specifically, Nilsson N. defined AI research as "*that activity devoted to making machines intelligent, and intelligence is that quality that enables an entity to function appropriately and with foresight in its environment*".[17] This largely differs from conventional computing, which is based on the principle of solving one computation at a time and solely for a given domain.[18]

AI is being deployed in both public and private sectors with profound implications for the economy and innovation.[19] It is a portfolio of fields; ML, CV, Robotics, Natural Language Processing (NLP) and more. Despite mass fear regarding high unemployment rates - to say the least -, AI has numerous advantages such as revealing patterns in high dimensional data, otherwise invisible to the naked eye. This enables growth in fields like drug discovery, highlighting our best chance in tackling humanity's greatest

enemies.[20] AI also revolutionises the way we communicate through voice recognition, natural language generation, machine translation and other.[21] Nevertheless, AI-powered robots can restlessly navigate and operate in any environment, relieving human workers from being deployed in remote or potentially hazardous workplaces.[22]

Rapid AI developments come after new ideas, guided by mathematics, the availability of big data and increasingly capable hardware. These are all contributing to the elevation of the technology, making it more 'accurate' in any considerable metric. A great paradigm is the acceleration of ML with the rise of quantum computing (QC). QC is expected to demolish the 'memory-wall' imposed by the use of a CPU or GPU, while quantum neural networks (QNN) allegedly train faster than any other known architecture.[23] In terms of 'fairness' and 'trustworthiness', explainable artificial intelligence (XAI) unlocks the potential for non-black-box algorithms which can be fine-tuned to mitigate bias. Research around these is part of many defense programs which aim to deploy autonomous battlefield robots.[24] Finally, 'prediction accuracy' is enhanced from new attention mechanisms, Bayesian optimisation techniques and other mathematical concepts.

2.1.1 Artificial Neural Networks

One of the most promising ML tools are neural networks (NN). They are fundamentally based on the universal function approximation theorem as presented by Cybenko in 1989.[25] They comprise of interconnected nodes, making them directly comparable to our perception of the biology of the human brain (where dendrites are connected with each other via routes called axons). Goodfellow I. et al. support that a feed-forward network with a single layer may represent any function, but the amount of neurons required may be too large leading to generalisation failure.[26] As such, it was suggested to position multiple layers in series to allow for increased representation capabilities, giving rise to a field known as deep learning (DL).

Each node of a layer has its own weights and biases assigned to it. The relationship

between the input x and output $f(x)$ of a NN with m layers is given by eq.(2.1).

$$f(x) = \sigma_m(w_m \times (\sigma_{m-1}(w_{m-1} \times (...\sigma_1(w_1 \times x + b_1)...)) + b_{m-1})) + b_m) \tag{2.1}$$

; where the output computation of each individual layer is an activated matrix multiplication between the layer's input and the layer's weights plus a bias. Different activation functions σ, techniques like dropout, batch normalisation and different operators such as convolutions can give different results for a wide range of applications.

The training procedure of a NN is a critical phase in its implementation. Its parameters θ must be optimised such that the given loss function L is minimised, eq.(2.2)

$$\theta_t^* = \underset{\theta}{\operatorname{argmin}}(L) \tag{2.2}$$

; where θ_t^* are the optimal parameters leading to the lowest possible loss value. L is usually a quantification of the discrepancy between the target and the output of the model, given a specific input, eq.(2.3)

$$L(x, \hat{x}) = \frac{1}{N} \times \sum_{i=0}^{N} (x - \hat{x})^2 \tag{2.3}$$

; where N is the number of training samples, x is the ground truth and $\hat{x}$ is the NN prediction. The optimisation is usually done using some form of backpropogation, fundamentally based on the chain rule and gradient descent, eq.(2.4)

$$\theta_i = \theta_{i-1} - lr \times \nabla_\theta L(x; \theta) \tag{2.4}$$

; where lr is the chosen learning rate and ∇_θ is the gradient of the loss with respect to the network weights. More on deep learning and the mathematical foundations behind it can be found in the Deep Learning book, Chapter 6.[26]

2.1.2 Physics-Informed Neural Networks

The optimisation algorithm presented above relies on vast amounts of data, such that the network parameters are optimised for as many example inputs x as possible. This makes NN less practical in applications where data acquisition is expensive, dangerous, or even impossible. The issue can be mitigated using additional information obtained from the enforcement of physical laws or any form of prior knowledge. Although the earliest related research dates back to 1997 from Lagaris I. et al., the idea did not receive much attention until 2019, when Karniadaks G. et al. revived and used it to present a deep learning framework for solving nonlinear PDE.[27, 28] The architecture was made known to the public as PINN and made use of mathematical systems and small quantities of data to collaboratively train a multi-layer perceptron.

Partial differential equations

A PDE describes the physical laws that govern the time-dependent responses of natural as well as human-made phenomena. In the spatial-temporal (X, t) domain, it has the general form as in eq.(2.5), constrained by boundary and initial conditions, eq.(2.6), (2.7)

$$u_t(X,t) = \Phi(u); \quad X \in \Omega,\ t \in [0,T] \tag{2.5}$$

$$B(u,X,t) = 0; \quad X \in \partial\Omega,\ t \in [0,T] \tag{2.6}$$

$$u(X,0) - I(X) = 0; \quad X \in \Omega \tag{2.7}$$

; where $u_t(X,t)$ is the solution of the PDE, $\Phi(u)$ is a non-linear operator and Ω is the computational spatial domain. A wide range of mathematical physics fall under this setup, such as conservation laws, kinetic equations, and most importantly for this study, diffusion processes.

Working principles

For the past decades, PDE were only partially solved using numerical simulation methods which need to divide the domain into finite elements (FE). On the other hand, PINN approximations are continuous; the laws of physics are embedded in the loss function, where the given constrains (interior, boundary and initial conditions) take the form of separate terms as seen in eq.(2.8)

$$L_{PINN} = w_f \times L_f + w_{BC} \times L_{BC} + w_{IC} \times L_{IC} + w_{data} \times L_{data} \tag{2.8}$$

; where L_f is the residual of the PDE, L_{BC} is the boundary residual, L_{IC} is the initial residual and L_{data} is the traditional discrepancy between the labels of the training data and the network predictions.

The residual f of the PDE can be seen as the subtraction of the r.h.s. of eq.(2.5) from its l.h.s. as in eq.(2.9).

$$f := u_t(X, t) - \Phi(u) \tag{2.9}$$

; where $u_t(X, t)$ is directly inferred using the main network for random proxy points in the given domain. On the other hand, the value of the non-linear function is obtained using GPU-accelerated automatic differentiation. Similar tactics are used for the initial and boundary conditions, depending on their type (Dirichlet, Neumann, etc.).

Simultaneous minimisation of all terms of the loss function is one of the main challenges. Different flavours of PINN have been used to tackle this issue, such as gradient-enhanced PINN (gPINN), where the gradient of the network itself is added in the loss as an additional term.[29] In addition, literature suggests that residual-based adaptive refinement (RAR), self-attention mechanisms and self-adaptive loss weights can theoretically and experimentally accelerate convergence in most cases.

Finally, Karniadakis G. et al. suggest that there is still need of further mathematical

understanding to produce robust and scalable next-generation PINN models.[30] For any given application, a rigorous analysis of the different design options is recommended, to understand their suitability and isolated effects. Current research focuses on the failure modes of PINN and new architectures such as spiking-PINN (SPINN).

2.2 Additive Manufacturing

AM, also known as three-dimensional (3D) printing, is a non-conventional manufacturing method based on a digital framework. It is regarded as non-conventional because it builds objects by adding layers upon layers of material, in contrast with conventional subtractive methods which remove material from a raw block. The technology is gaining popularity due its rapid and cost-effective prototyping capabilities, the expected decrease in produced pollutants (emissions, material waste) and its potential to manufacture previously infeasible complex geometries.[1] Such geometry is a hollow part, up to 63% lighter and equally strong to its solid counterpart.[31] As such, AM reduces product development costs while increasing performance. Revolutionary related inventions include developing an artificial human heart from Cohrs et al.[32] Meanwhile, the space industry uses the technology to integrate and print various advanced rocket parts. There are also speculations that deploying large-scale AM machines on Mars to autonomously fabricate habitable infrastructure is a viable solution.

Research around AM covers many aspects. Recent improvements include 4D printing, initially introduced in 2013 as the advanced version of 3D printing. Using it, intelligent materials can be printed which change properties due to the influence of outside energy inputs. For example, electro-active polymers (EAP), exhibit change in size when stimulated by an electric field.[33] Other smart materials have self-repairing capabilities.[34] These are being used in state-of-art bionic mechanisms, such as compliant passively actuated joints. In addition, micro-extrusion gave rise to additive micro-manufacturing, a

promising technology for medicine and drug delivery. It is capable of manufacturing parts ranging from 0.5 mm to 2 mm with a resolution of 5 µm to 20 µm.[35] Hybrid processes have also been introduced, giving the possibility of combining all aforementioned methods with AM-chemical treatments to produce multi-property parts.[36] The US Air Force is using this to integrate printed sensors (e.g., strain gauges and thermistors) within parts, used during testing and deployment to assess quality and performance.[37]

Besides the increase in scientific understanding and development, the 3D printing market value was estimated at USD 13.7 billion in 2020. During the COVID-19 pandemic, it managed to develop healthily and remained one of the most active markets globally. It is expected that it will reach USD 63 billion by 2026. The US has recently launched the 'AM Forward' initiative, aiming to improve its supply chain resilience by funding related research centers. Reports have brought to the forefront a future where the average household is equipped with at least one related device.[38] As such, the hype around the technology is expected to rise, flexing its potential to current and new investors.

Despite the development, there are still challenges to be addressed. The most important, from an engineer's point of view, is most additive processes suffer from low quality consistency in terms of geometric dimensioning and tolerating.[39] This is commonly attributed to non-deterministic variables of the printing environment, reducing the reliability of AM while also discouraging companies from using it in large-scale risk-averse supply chains. Another underlying challenge is the risk imposed from the inhalation of evaporated particulate matter released during the involved thermal processes. Research supports that these emissions can be significantly reduced if specific printing parameters are adjusted and can be up to 97% mitigated using EPA filters.[40] This is relevant to FDM which deals with the extrusion of thermoplastics and PBF that uses directed deposited energy (high-power laser or electron beam) to fuse material powder. Lastly, governments and the regulatory authorities are puzzled by a more complex issue; the technology's capability

to manufacture virtually anything, gives its users the freedom to get hands-on restricted objects such as firearms. Before establishing AM as a reliable and safe manufacturing method, these must be addressed.

2.2.1 Laser Powder Bed Fusion

Although numerous categories of materials can be printed with different additive methods, this project focuses on the LPBF of metal. AM will hereafter refer to this. In LPBF, a thin layer of material powder is added on the working platform, before a high-power laser beam is projected on it with a CAD-informed trajectory. Any powder grains subject to radiation will undergo a phase change from solid to liquid; during re-solidification the melted material will fuse together, while the rest remains in powder form. This process is referred to as a 'layer scan'. When a layer scan finishes, the build platform is lowered by a height equal to a pre-specified layer thickness and new powder is added above; once again, a CAD-informed layer scan accumulates solid material at designated areas on the 2D plane. The process continues until all layers of a sliced object are processed to form the required 3D geometry. The overall setup of an LPBF machine is visualised in Figure 2.1, along a close-up of the laser-matter interaction, both adapted by [41].

Arguably, the most important factor is the laser, pulsing at >10 kHz. That's 36 million spot welds per printing hour, with a melt pool diameter no larger than 1 mm each. This highlights the macroscopic and microscopic scale in which AM is happening. Further details regarding the involved physics and dynamics can be found in [41].

2.2.2 Part Qualification

The concept of qualification is defined as the process of gathering evidence that a part meets certain requirements, such as minimum or maximum weight and adequate strength. The scope is to confirm that the finished part will be able to operate at certain standards

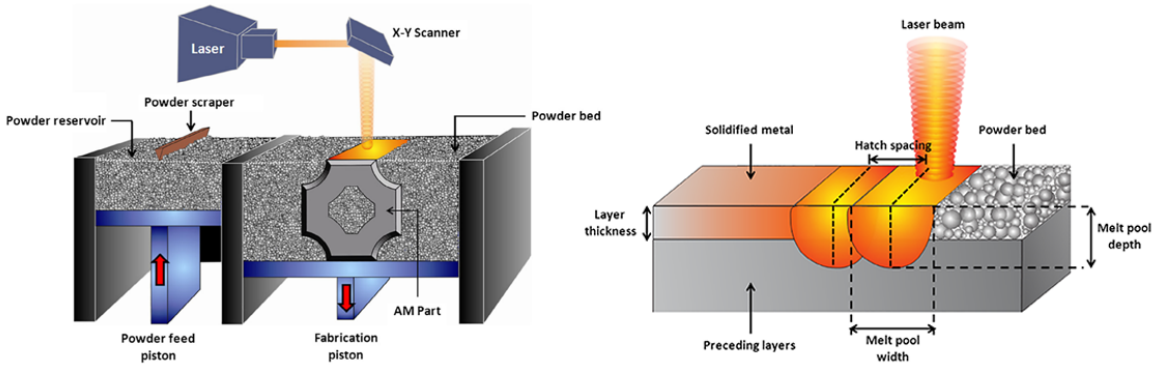


Figure 2.1: Schematic representation of laser powder bed fusion process setup (left) and close-up of the laser-matter interaction (right).

if deployed. Companies typically develop and employ internal standards (e.g., see NASA Engineering and Quality Standard for Additively Manufactured Spaceflight Hardware) to avoid the bureaucracy required before the authorities publish external ones.

The qualification procedure becomes harder when part-to-part variation is taken into account. This refers to the observation that parts printed from the very same machines have been found to be inconsistent on the micro-scale. Such inconsistencies are mostly related to melt pool instabilities and come in the form of porosity, variations in level of fusion between layers and micro-cracks induced from residual stress. Residual stress occurs due to high cooling rates and steep thermal gradients. In general, defects within a part's volume act as stress concentrators that can be detrimental to performance and fatigue life.[42] To achieve qualification, parts must be monitored and analysed on an individual basis.

Conventional qualification procedures exist, however, in the context of AM they are considered less than ideal. This is because they are either destructive or can only provide an ex-situ, i.e., post-build, analysis. On one hand, destructive methods defy the purpose of qualification; destroying a part to confirm its quality makes it unusable and any conclusions can't be used for other prints due to part-to-part variation. On the other hand, ex-situ methods such as X-ray computed tomography (XCT) or laser ultrasonics are more useful but still limited when dealing with complex geometries. Neutron diffraction has also been used ex-situ; it allows greater penetration than X-rays, however, there is scarce

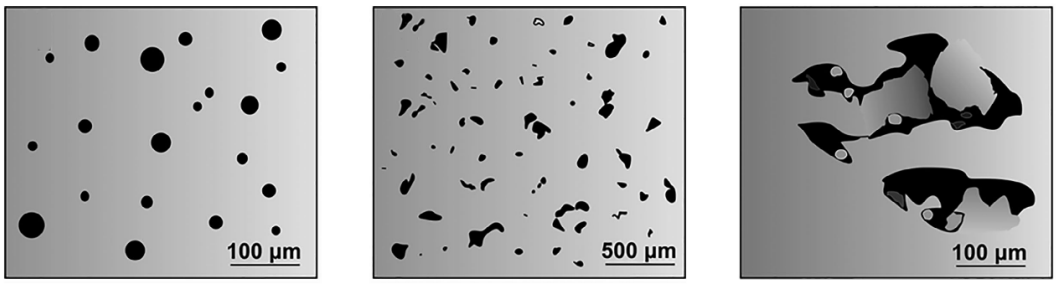


Figure 2.2: Typical defects in laser powder bed fusion: gas-induced (left), keyhole (centre) and lack of fusion (right) porosity.

availability of neutron sources. Another more obvious method is testing, however the procedure can become lengthy if setup is required. Testing can also threaten the structural integrity of expensive equipment in the case of failure.

Engineers have drifted towards real-time in-situ qualification methods, to develop controllers that can autonomously adjust process parameters to enforce melt pool stability. In addition there are hopes that errors can be actively repaired based on the quality mappings of preceding layers. This has as a prerequisite accurate localisation of defects. Generally, porous regions can be remelted such that trapped gas is released while regions with weak fusion can be reinforced. Luckily, the latter is often observed even without adjustments, a phenomenon known as 'self-healing'.[43]

Quality issues

Different failure modes can cause different defects. The modes are directly related to powder spreading issues and sub-optimal laser-powder interaction. Below is a summary of the basic LPBF defect types; gas porosity, keyhole porosity and lack-of-fusion, with comments on their severity. Figure 2.2, adapted by [41], visualises their differences in terms of size and geometry.

Gas pores, also called metallurgical pores, occur from the inclusion of gas in the volume during the phase-change reaction.[41] They are highly associated with welding processes where gas sources are abundant, however, in LPBF the working environment is fully

protected from an inert atmosphere. As such, the only remaining gas sources in the chamber are the shielding gas and the powder particles themselves; research supports that careful selection of process parameters and correct powder preparation procedures can effectively minimise their formation.[44] Gas pores are usually spherical and smaller than 5 microns in diameter; when in normal concentrations they have minimal impact on the part's quality.

Keyhole pores contribute most of the porosity formation in LPBF and are mostly caused due to excess energy input. As the energy input is a function of laser power and velocity, their presence becomes particularly notable when these variables take extreme values. Such regions are turning points; the laser's velocity decreases and eventually reaches 0 m/s.[45] While static, the heat input at the turn point increases linearly with time, leading to excessive melting and vaporisation of the material in the region. In parallel, the radiation pressure arising from the momentum of the incident photons penetrates the surface of the melted material and forms cavities. As the laser starts moving again towards the opposite direction, the high cooling rates 'freeze' these cavities before the viscous force anticipates them. As such, large quantities of gas are trapped in the solidified volume. This acts as a catalyst for the part's surface roughness. Keyhole pores are typically larger than 100 microns in diameter, can occur at large depths from the working surface, and constitute the most severe porosity type in LPBF.

Lack of fusion defects, as the name suggests, refer to voids between layers or scan tracks due to insufficient penetration or melting, owing to weak laser-material interactions.[9] Such interactions are characterised by inadequate laser energy input. In real cases, even if nominally adequate laser power is used, the generation of plume can result in the attenuation of the beam. Powder dislocation and spatter re-deposition may also lead to an uneven thickness of the powder layers or partial obscuration of the beam, ultimately deteriorating the melting process. As mentioned above, these are often self-repaired during subsequent layer scans. Lack of fusion pores are up to 500 microns in length and have a characteristic grain-like shape.

2.2.3 Digital Twins

A DT is defined as the virtual representation of a dynamic system that spans its life-cycle, updated in real-time to help decision-making. The concept was born at NASA in the 1970, to evaluate the causes of Apollo 13's oxygen tank explosion.[46] Nowadays, such frameworks are used in every engineering sector; manufacturing, robotics, medicine, etc.

In the context of AM, a DT could assist in optimising process parameters, improving part quality via monitoring of process flaws, and shortening the time to qualify printed parts. According to DebRoy et al., this is to be done using a framework that consists of mechanistic and statistical models, fused sensor data and machine learning.[47] Mechanistic models represent our knowledge of the physical systems surrounding the involved process. Sensor data will bridge reality and the digital world, and can be captured using cameras of any type, acoustic and vibration sensors, pyrometers and other. Machine learning will mimic our intellect by utilising algorithms that have been widely researched by researchers in practically all domains.

One of the bottlenecks is that the use of mechanistic models is highly dependent on simulations. These are traditionally solved using numerical analysis techniques, based on breaking down the given domain in small discrete elements. As the assumed resolution increases, the size of the mesh increases proportionally, adding to the complexity of the system. One of the most widely used numerical solvers is FDM which approximates the derivatives of the given system using finite differences. More on numerical analysis techniques can be found in [48].

Conclusion

The essential AM and AI principles employed in this work have been explored. A preliminary attempt to classify and localise defects has also shown that purely data-driven ML algorithms are not highly effective (see Appendix: Porosity classification). This has

been attributed to difficulties arising from labeling the available data and the inability of the employed methods to assimilate physical laws. DT aim to incorporate mechanistic models in the form of physical simulations to improve this. However, it is explained that numerically solved simulations lack in terms of time and space complexity. PINN, a new class of DNN, have shown immense potential in solving PDE systems with a number of advantages when compared to traditional approaches. For one, they provide mesh-free solutions in the continuous realm. They can also train without training data and in a self-supervised manner. Their memory requirements is negligible and directly proportional to the amount of NN parameters used. However, because they are relatively new, their behaviour has yet to be extensively examined.

Chapter 3

Methods and Experiments

All methods and experiments prepared for this project are presented in this chapter. The underlying principles and ideas to overcome the challenges are explained, and isolated performances of different implementations are reported. A summary of the conducted experiments can also be found in Table 3.1.

Table 3.1: Summary of conducted experiments.

Ref.	Description
Vanilla-FDM	Implement a finite difference method solver to simulate a single-track deposition experiment at constant laser settings.
Vanilla-High-Fid	Implement a finite difference method solver to simulate a high-fidelity deposition experiment with dynamic laser settings. To incorporate real-world dynamic changes, data obtained from a co-axial imaging system are used.
PINN-ENA	Implement a physics-informed neural network to simulate a single-track deposition experiment at constant laser settings.
PINN-DYO	Implement a physics-informed neural network to simulate a high-fidelity deposition experiment with dynamic laser settings.
PINN-TRIA	Implement a parameterised physics-informed solver to simulate the single-track deposition experiment over a combination of laser settings and wrap them in a single neural network.

3.1 Data Preparation

While most subtractive processes have their own well-established in-situ monitoring methods (e.g. 'REALISM' by CORDIS EU), no out-of-the-box AM machine incorporates similar technologies to enable discontinuity detection in real-time. Subsequently, different research groups attempt to tackle the gap in a variety of ways, and diverse datasets in terms of structure, size and quality are constantly being generated. The data engineering practices followed for this report are discussed below.

Data collection

A high-speed melt pool intensity imaging system has been utilised for data collection. The system was installed in a Renishaw AM250 metal LPBF machine, allowing the imaging of the melt pool region along a path co-axial to the laser beam path. The co-axial system was considered a better solution when compared to other imaging options, mainly because it enabled the capture of the melt pool spot in a Lagrangian frame of reference. The region of interest remains static w.r.t. the camera's field of view, although the region of interest is actually moving w.r.t. the camera's physical position. This is achieved via a sophisticated optics pipeline, Figure 3.1.

In sync with the captured image stream, the in-situ monitoring system records the laser's position, power and focus. The velocity can also be calculated using the implemented turning point detector (see Appendix). This set of values will be referred to as the 'metadata'. More on the imaging system can be found in [49].

Feature extraction

The video frames were converted into physically meaningful descriptors. This was done to minimise the memory size of the files involved, resulting in greater portability. Furthermore, reducing the data dimensionality from 128x128 for each frame to a short 1D

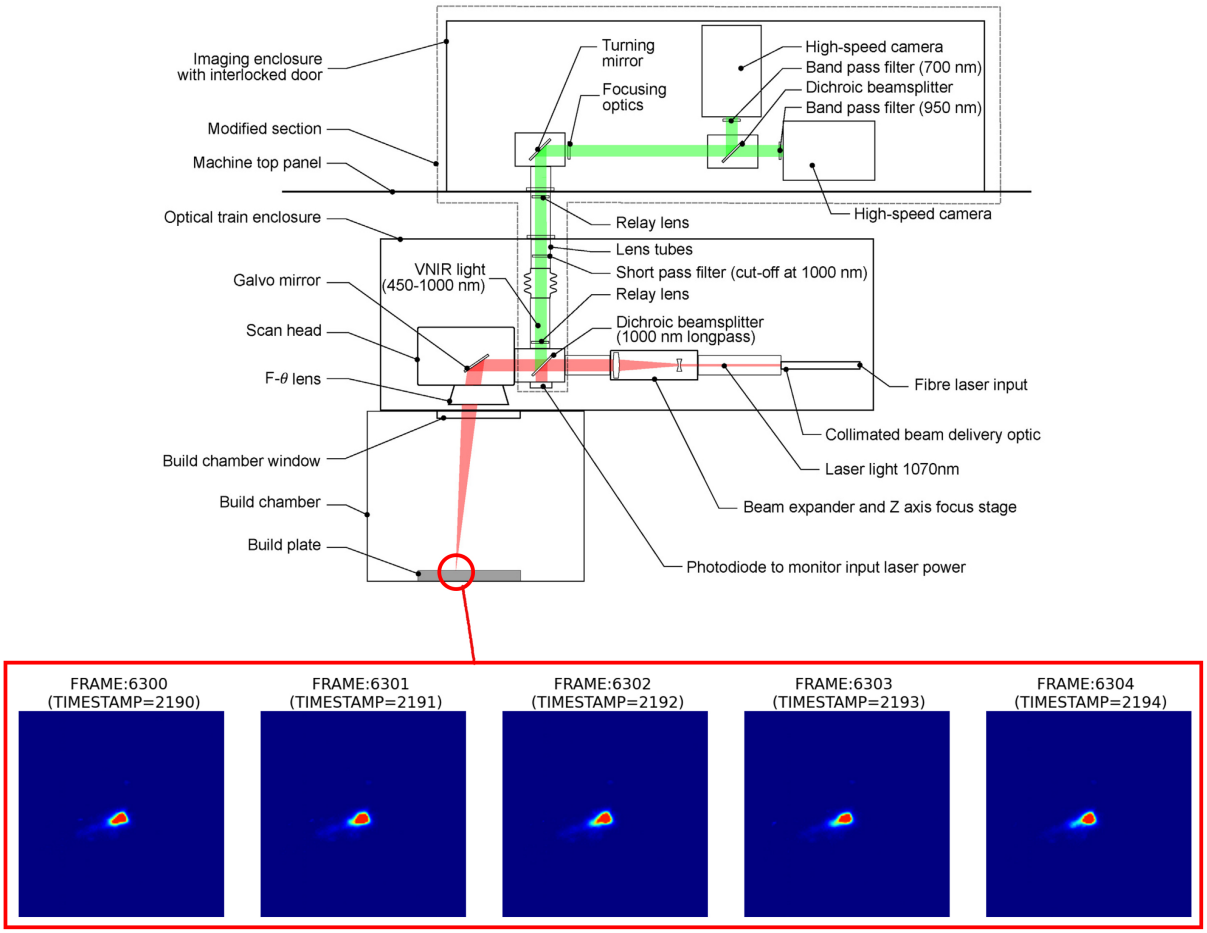


Figure 3.1: Layout of the commercial laser powder bed fusion machine and modification to allow coaxial imaging of the melt pool (top) and sample sequence from 20 kHz high-speed cameras (bottom).

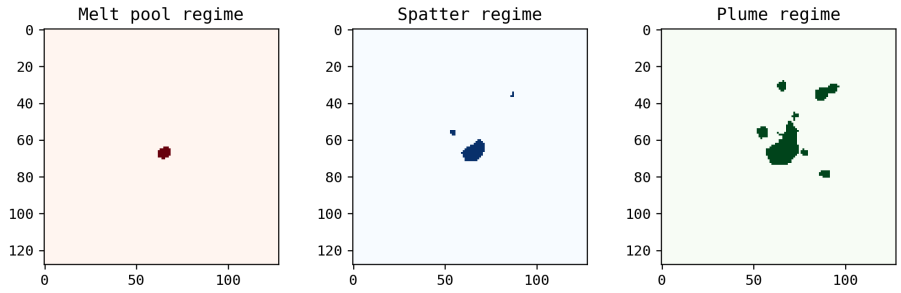


Figure 3.2: Three-channel segmentation of sample frame from optical data stream. Activation thresholds are >240 for the melt pool (left), >150 for the spatter (centre) and >10 for the plume (right) regimes.

Table 3.2: List of physically meaningful features extracted from optical data stream.

Melt pool area	Melt pool mean intensity	Melt pool roundness
Spatter quantity	Spatter total area	Spatter major area
Plume total area	Plume major area	Plume roundness

array boosted the ability for real-time processing, a difficult challenge given the high data collection rates of 1 terabyte of data per printing hour. In addition, Smoqi Z. et al. suggest that using physically intuitive process signatures in simple ML models is as effective as using a black-box DL approach.[50]

The feature extraction was performed using a three-stage automatic pipeline. In stage A, the frames were segmented into 3 channels using threshold filters, Figure 3.2. The threshold values were strategically chosen via trial and error to visually distinguish between plume, spatter, melt pool spot, and background regimes. In stage B, each channel was analysed using skikit's *measure.label* function, which detects neighbouring activated pixels and turns them into clusters. In stage C, a number of characteristics, such as cluster area, centroid, perimeter, and so on, were obtained by processing the structure of each cluster using scikit's *measure.regionprops* function.

This study makes use of the melt pool spot's area and the plume's mean intensity. Melt pool area is defined as the area of the largest cluster detected in the melt pool regime channel. Plume intensity is defined as the mean intensity of all pixels activated in the plume regime channel, minus the intensities from the pixels activated in the area and spatter regime channels. However, the concept was expanded to provide a total of 9 features per frame, Table 3.2, for future work. It is also suggested that a CNN can be trained in a self-supervised manner to extract features with less computational expense.

3.2 Thermal Modeling with Numerical Methods

In this section the heat diffusion problem is defined in the context of AM. It is then solved using numerical methods for experiments involving different settings and levels of fidelity. The results are not explained because they are solely used to validate the state-of-the-art solvers introduced in the following section.

3.2.1 Heat Diffusion in LPBF

Heat transfer is a fundamental component of the world around us. It is defined as the branch of thermal engineering dealing with the generation, conversion, and exchange of thermal energy across physical systems. Diffusion, in general, refers to the spread of a quantity from high to low concentration regions. In AM this concept is particularly important, as the method itself heavily relies on energy inputs from the laser beam which travels through the inert gas before reaching the powder layer and the solid geometry beneath.

Specifically, the 3D heat diffusion PDE is the system that characterises the variation of temperature in a spatial domain over time, eq.(3.1)

$$\left(\frac{\partial T}{\partial t}\right) \times \rho \times c_p = \frac{\partial}{\partial x}\left(k\,\frac{\partial T}{\partial x}\right) + \frac{\partial}{\partial y}\left(k\,\frac{\partial T}{\partial y}\right) + \frac{\partial}{\partial z}\left(k\,\frac{\partial T}{\partial z}\right) + \dot{q}_{laser} \tag{3.1}$$

; where T is the temperature (K), c_p is the heat capacity (J/gK), k is the heat conductivity (W/m^2K) and ρ is the specific density (kg/m^3) of the material. $\dot{q}_{laser}$ is the volumetric heat generation (W/m^3) caused by the laser, specific to the LPBF setup and further discussed below.

A visualisation of the assumed spatial domain is in Figure 3.3. In terms of constraints, Dirichlet conditions are assumed for the initial conditions ($t = 0$), and Neumann conditions are assumed on the boundaries of the domain. All sides of the domain are treated

as isolated surfaces, except the bottom surface ($x =:, y =:, z = 0$) from which heat freely propagates to the preceding layers.

The power loss from convection and radiation can be included in the system with the addition of the $\dot{q}_{conv}$ and $\dot{q}_{rad}$ terms in the boundary constraints of the system. Their effects are beyond the scope of this study and thus have been neglected. This simplified version of the system also fails to account for individual powder particles, as well as other phenomena like molten material flow. Including these would require advanced numerical simulations usually relying on CFD. The system does however take into consideration variations from the thermal-dependent material properties, k and c_p. The assumed material is 316L stainless-steel which melts at 1375 K and has specific density equal to 7966 kg/m^3.

Heat source

Literature for AM simulations, highly influenced from that of welding, has used a wide range of different moving heat sources.[51] The two main types are 2D sources which describe the energy input per unit of area per unit of time (W/m^2) and 3D sources that describe the energy input per unit of volume per unit of time (W/m^3). On a technical note, using the former is known as applying heat flux boundary conditions on the sides of the domain, whereas the latter is known as internal volumetric heat generation because the source directly interferes with the interior state of the domain.

Zhang et al. support that 2D sources fail to account for the heat source penetration, an important factor when dealing with the fusion between layers or the keyhole melting mode.[52] They also reviewed and validated eight different volumetric heat source models; the semi-spherical heat source model was selected for this study, eq.(3.2)

$$\dot{q}_{laser} = \frac{10.39 \times P}{5.57 \times r^3} \times exp\left(-3 \times \frac{(x - v \times t)^2 + y^2}{r^2}\right) \times exp\left(-3 \times \frac{z^2}{c^2}\right) \tag{3.2}$$

; where P is the laser's power, r is the laser's focus radius, v is the laser's scan speed and c is the laser's penetration depth. It was consider suitable due to its relevance with the

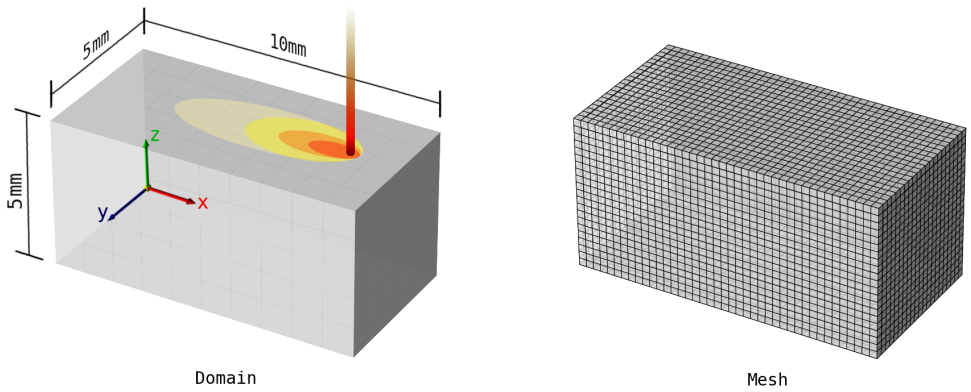


Figure 3.3: Visualisation of the 3D domain (left) and 41x21x21 example mesh (right) assumed during thermal modelling.

widely recognised Goldak's double-ellipsoid heat source model with a substantial decrease in implementation complexity.[53]

Because the laser's penetration depth can not be directly extracted from the available data, it was assumed constant and equal to x2 the powder layer height (=50 µm) in all experiments. This value was selected as the incident light is expected to reflect between powder particles until it reaches - and slightly penetrates - the surface of the preceding layer. This value may be useful in calibrating the source to represent real-world operation settings in future work.

Table 3.3: Laser settings used for different single-track deposition experiments.

Bench	Laser Settings			
(ref.#)	Power (W)	Speed (mm/s)	Focus (mm)	Depth (mm)
1	200.0	200.0	0.50	0.05
2	182.0	305.0	0.23	0.05
3	203.0	217.0	0.31	0.05
4	218.0	510.0	0.21	0.05
5	193.0	420.0	0.28	0.05

3.2.2 Single-Track Deposition

To numerically solve the formulated problem, an FDM simulation was scripted using Python. It was programmed to calculate the solution of the system at discrete elements defined in an (X, Y, Z, T) mesh, Figure 3.3, using the Forward in Time - Centered in Space (FTCS) scheme. The laser settings used can be found in Table 3.3. Multiple simulations were run with different settings such that their solutions can be used as ground truth during the benchmark evaluation of succeeding experiments.

The simulation duration was carefully set such that the source has adequate time to travel from x_0 to x_1, even for the lowest assumed scanning speed. This distance was chosen such that the induced temperature profiles have enough time to stabilise, happening after 1.5 mm from the start of the track.[54] As such, for a 10 mm stroke at 100 mm/s, in 0.1 seconds the solution can capture the transient and steady thermal states of the melt pool and the domain's cooling behaviour. For reference, 'transient thermal states' refers to the thermal profiles observed while the melt pool is still being formed.

The minimum time quantisation was calculated using the Courant-Friedrichs-Lewy convergence condition, eq.(3.3)

$$\Delta t = C \times (\Delta x)^{-1} \tag{3.3}$$

; where C is the Courant number. When C is smaller than one, stability is guaranteed throughout the solution. Other factors considered for selecting Δt were the length of the laser pulse and literature suggesting that finer time quantisation is needed to integrate the temperature-dependent material characteristics.

It should be noted that from 3.3, a finer spatial mesh leads to higher solving times, backed by Table 3.4. Although optimised FDM simulations run much faster via parallel calculations and graph theory, their time complexity is still widely regarded as high. The generated solutions need also to be explicitly stored; that is, for a low/medium mesh quality, 217e+6 floats with a total memory requirement of 81.92 Mb. Due to the used ma-

Table 3.4: Mesh configurations used to solve presented problem with finite difference method.

Mesh Settings				Solver	Solution
x	y	z	t	time (s)	size (Mb)
41	21	21	800	194.217	5.12
81	41	41	1600	3003.634	81.92
101	51	51	2000	13121.312	512.28
121	61	61	2400	failed	failed

chine's active memory limitations, the last simulation, which employed a 121x61x61x2400 mesh, failed.

In total, solving the FDM for all 5 referenced setting combinations took 13h 12m for a 101x51x51x2000 mesh. A video has been prepared to visualise the FDM results of the single-track experiment (ref.#1) and is available at Vanilla FDM.

3.2.3 High-Fidelity Deposition

The single-track experiment is an idealised scenario; in real life, some machine parameters fluctuate due to design requirements and stochastic or systematic errors. For example, the laser's power is not constant and its velocity is a function of the desired path. More importantly, the melt pool does not have a constant shape for factors such as variability in the laser-matter interaction due to the presence of plume. To provide a thermal analysis more closely aligned with reality, these variations must be considered.

Using the available co-axial imaging system, parameter values at a given time can be extracted either directly from the available metadata or indirectly from the extracted descriptors. A random 200-long sequence (of length 10 milliseconds given 20 kHz collection rate) of such values was extracted from layer 103 of the recorded build. Their evolution across time can be seen in Figures 3.4 and 3.5. To account for their dynamic changes during the numerical analysis, the original FDM script was modified to switch its parameters

to the appropriate time-dependent parameter-set at each iterative computation.

One of the non-trivial tasks was to account for the non-straight path of the laser. On the same note, the number of known laser positions is 200 but the number of time-steps occurring from the time quantisation is much higher. For high resolution results, the path had to be interpolated such that the laser position slightly changes at all processed times. Using the known laser locations and velocities, intermediate positions (l_x, l_y) were calculated as can be seen in Figure 3.6. These calculated positions were directly used in the slightly modified source equation, eq.(3.4)

$$\dot{q}_{laser} = \frac{10.39 \times P(t)}{5.57 \times r^3} \times exp\left(-3 \times \frac{(x - l_x(t))^2 + (y - l_y(t))^2}{r^2}\right) \times exp\left(-3 \times \frac{z^2}{c^2}\right) \quad (3.4)$$

; where $P(t)$, $l_x(t)$, $l_y(t)$ are the time-dependent laser power and offsets. It should be emphasised that the laser is continually moving, while other parameters have to remain constant between known locations. To improve this, a faster monitoring system could be used to sample ground truth at a higher rate.

The plume intensity was finally neglected from the analysis. Upon research and feedback relating to the laser-plume interaction, it was noted that the main effects on the beam are attenuation and de-focusing.[55] However, quantifying these effects - even more so accounting for them in a simulation - is no trivial task. A method to elaborate on the laser-plume relationship is suggested in Chapter 5.

The high-fidelity simulation took 1980.111 seconds to solve for a mesh consisting of 81x41x41x320 nodes. Other mesh sizes were not considered because the complexity trends identified in the single-track experiment are likewise predicted here. A video was prepared to visualise the FDM results of the high-fidelity experiment and is available at High-fidelity FDM.

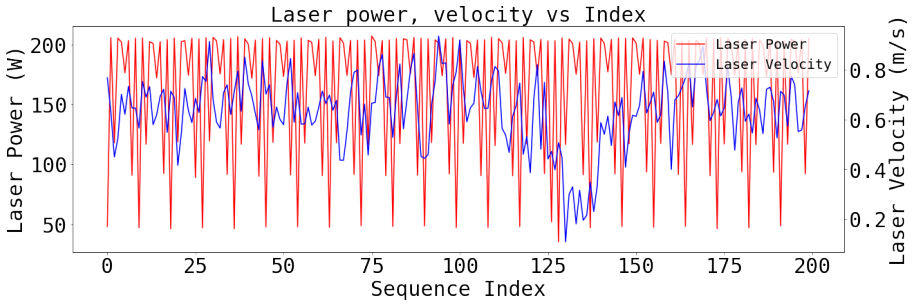


Figure 3.4: Evolution of laser power (W) and velocity (m/s) extracted from metadata of optical data stream.

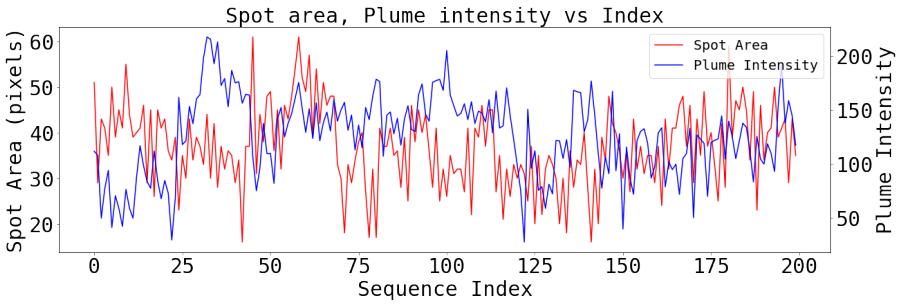


Figure 3.5: Evolution of melt pool spot area (pixels) and plume mean intensity extracted from optical data stream.

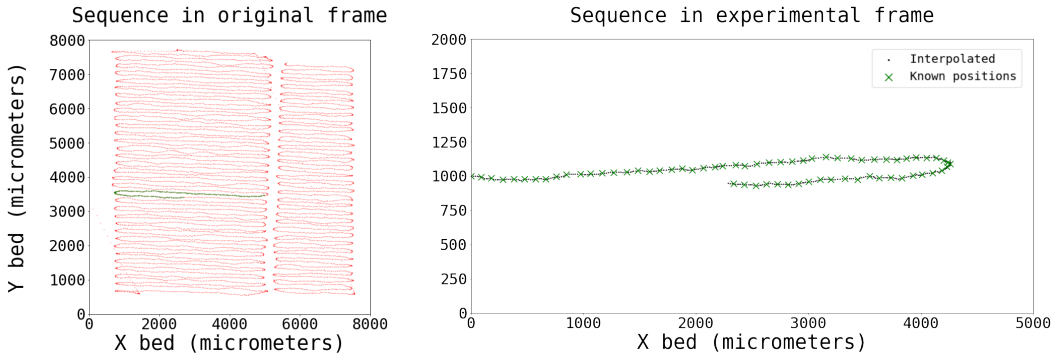


Figure 3.6: Sampled sequence in original frame of reference (left) and transformed sampled sequence superimposed on completed path in experimental frame of reference (right).

3.3 Thermal Modelling with PINN

It has been shown that modelling AM processes with finite methods can become a lengthy and intense computational task, especially when higher levels of fidelity are required. Furthermore, heat diffusion is a forward problem, which implies that if a parameter changes, the FDM solution must be re-calculated starting from t_0. As described in the background section, PINN have previously exhibited capability in approximating any PDE using automatic differentiation with multiple benefits when compared to classic approaches. In this section, a basic PINN is first implemented to solve the single-track deposition experiment. The high-fidelity experiment is then replicated using a hybrid data-driven approach. Finally and most importantly, a parameterised solution for different heat source settings is 'wrapped' in a single network.

3.3.1 PINN-ENA: Implementation and Tuning

In this section a PINN is implemented and tailored for the single-track deposition experiment. Its overall schematic can be seen in Figure 3.7. After defying the main network, residual network and implementing the loss function, a working surrogate was achieved. Different hyper-parameters and design choices (input and output normalisation, activation function, optimiser, main network architecture) are then assessed to increase prediction accuracy and convergence performance. All of the information obtained about the network's behaviour is utilised in subsequent experiments.

A short video was prepared to visualise the PINN results of the straight-track deposition experiment and is available at Vanilla PINN. The evolution across time is slowed down to highlight the capability of the network to generalise for any time t.

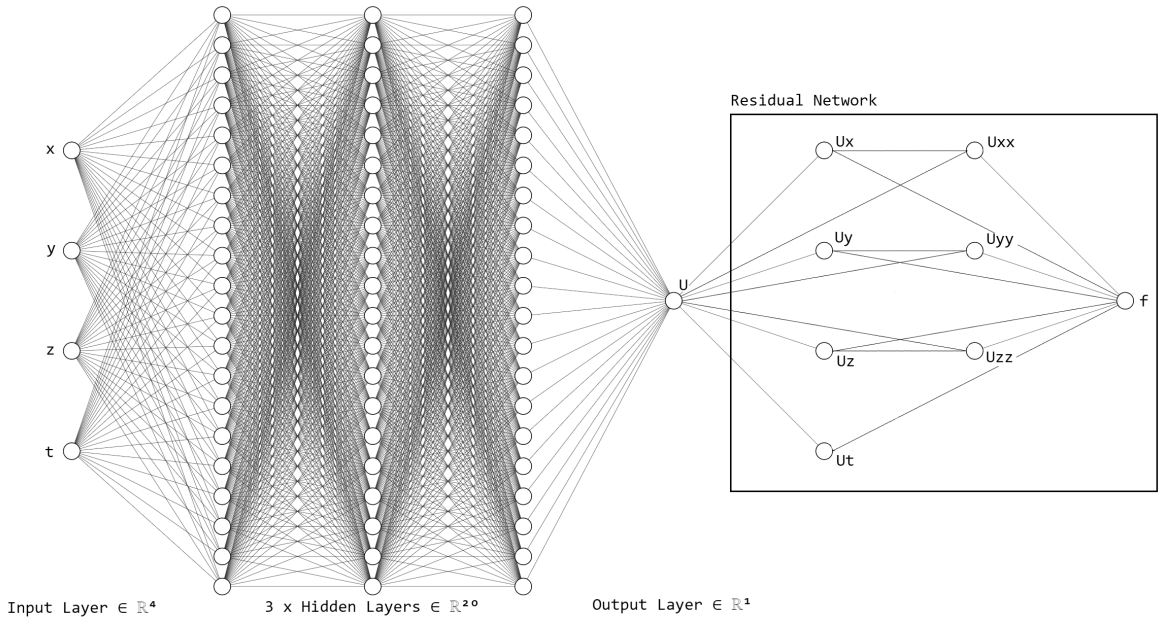


Figure 3.7: Schematics of physics-informed neural network used for experiment PINN-ENA, consisting of 3 hidden layers with 20 neurons each. Residual (f) network is based on automatic differentiation powered by PyTorch, where Ux, Uxx, Uy,... are first and second partial derivatives of the main network's output U w.r.t. its inputs (x,y,z,t).

Training proxies

PINNs have the flexibility of being trained in a self-supervised manner without the need of ground-truth training data. However, training proxies are still required for the calculation of the residuals used during optimisation. Each point set: co-location (interior), initial and boundary proxies, must have certain distributions tailored to the problem formulation, Figure 3.8. In any case, maximising their density is desired, to help the network generalise. However, more proxies come at a cost so efficient sampling methods are required.

The most simple sampling method is uniform sampling, which positions proxies in the domain in equally spaced intervals. It can be easily implemented using a combination of Numpy's *linspace* and *meshgrid* functions. This tactic has been previously used by Shuheng et al., however, it is suggested that placing training proxies in arbitrary positions will help for better convergence.[15] In addition, it is suggested that different proxies should be used at each time-step, to increase their overall density within the domain.

To generate proxies with arbitrary positions, Latin hypercube sampling (LHS) is used.

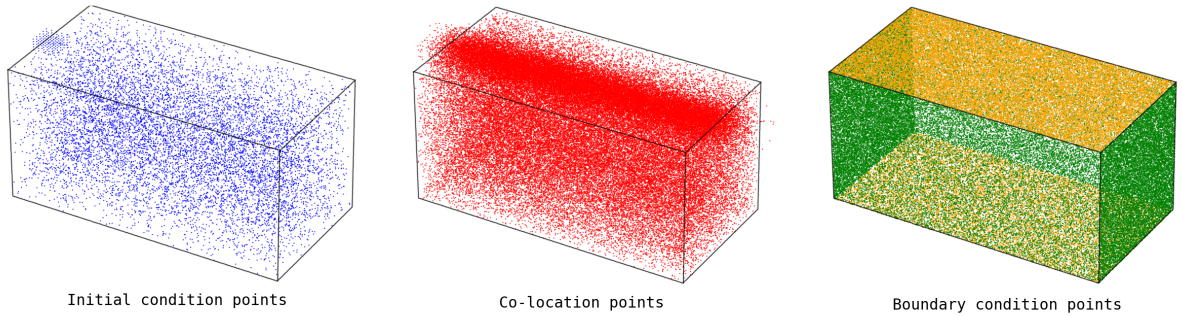


Figure 3.8: Training proxies generated in spatio-temporal domain using Latin hypercube sampling. Initial condition points (left) are distributed in spatial domain for $t = 0$. Co-location points (centre) are distributed in spatial domain for $t \in [t_0, t_{max}]$. Boundary condition points (right) are distributed on the physical boundaries of the spatial domain for $t \in [t_0, t_{max}]$.

This is achieved using pyDOE's *lhs* function which takes as input the space dimensionality and the amount of points to be sampled. The elements of the original output D_{LHS} are normalised; to fit the generated distribution on the problem's dimensions, eq.(3.5) is used

$$D_{domain} = X_{min} + (X_{max} - X_{min}) \times D_{LHS} \tag{3.5}$$

; where D_{domain} is the fit distribution, X_{min} and X_{max} are the lower and upper boundaries of the domain and D_{LHS} is the original normalised LHS distribution.

One of the important structural characteristics of the co-location proxies is higher density near the moving source position, visible in Figure 3.8. This was done in accordance with literature suggesting that more co-location points should be positioned in regions where the gradients of the system are expected to be relatively steep. To achieve this, a 3D Gaussian distribution is dynamically sampled near the location of the source at each assumed time-step, then superimposed on D_{domain}. The scale of the secondary distribution affects the convergence and accuracy performance; the best balance was observed when 95% of the points fall within x2 the laser's diameter.

During implementation the number of training proxies used was 1e+5, 6e+4 and 1e+4 for co-location, boundary and initial respectively. Increasing the amount of proxies imposed unnecessary workload on the GPU and increased the convergence time.

Residual network

The residual network is responsible for quantifying the violation of the PDE at the proxies. To do this, each proxy (x, y, z, t) is input into the main network for a prediction u to be made. Using PyTorch's automatic differentiation, the output is partially differentiated w.r.t. its inputs and the obtained gradient graphs ($\frac{\partial u}{\partial t}, \frac{\partial u}{\partial x}, \frac{\partial u}{\partial y}, \frac{\partial u}{\partial z}, \frac{\partial^2 u}{\partial x^2}, \frac{\partial^2 u}{\partial y^2}, \frac{\partial^2 u}{\partial z^2}$) are used in eq.(2.9). The network's weights are then penalised such that the residual is minimised.

The same principles are used to implement the boundary and initial conditions, however, separate enforcing pipelines are needed for each.The former trains the PINN so that the first derivatives of its outputs w.r.t. its inputs is 0 near the domain's physical limits - this is identical to Neumann boundary conditions. The latter trains the PINN so that whenever the input feature t is equal to 0, the output u at any location (x, y, z) takes a specific value - this is identical to Dirichlet boundary conditions.

To integrate the training proxies and the three enforcers, 'flags' were used. One flag was assigned to each generated point-set; 8 flags were assigned in total ('DOM', 'IC', 'X0', 'X1', 'Y0', 'Y1', 'Z0' and 'Z1'). The first two were used for the interior and initial proxies while the remaining six were used for the boundary conditions at the different sides of the domain. The boundary proxies are separated, such that the Neumann condition is applied in the appropriate direction. During each training iteration, as the 8 different point-sets are handled, the flags inform the trainer which part of the domain they cover, and as such, which enforcer they should be used in.

Loss function

The three residuals are calculated and then used in the loss function, eq.(3.6)

$$L = \lambda_{Res} \times L_{Res} + \lambda_{BC} \times L_{BC} + \lambda_{IC} \times L_{IC} \tag{3.6}$$

; where L_{Res}, L_{BC} and L_{IC} are the interior, boundary and initial conditions residuals

$$L_{Res} = \frac{1}{N_{Res}} \times \sum |\hat{u}(x, y, z, t)|^2 \tag{3.7}$$

$$L_{BC} = \frac{1}{N_{BC}} \times \sum |\beta(\hat{u}'(x, y, z, t))|^2 \tag{3.8}$$

$$L_{IC} = \frac{1}{N_{BC}} \times \sum |\hat{u}(x, y, z, 0)|^2 \tag{3.9}$$

The λ values aim to provide a good balance between the different terms during training. Intuitively, the interior residual was initially weighted more than the other two. This was done because the heat generation term is embedded in it, considered to be the most important available knowledge. Upon experimentation it was observed that in the naive case of constant material properties this had little to no effect. However, the assumption was correct when the thermal-dependent material properties were introduced. When the lambda weights were not tuned either of the following scenarios occurred: i) the interior and boundary residuals drifted to extreme values while the network solely minimised the interior residual or ii) the interior and boundary residuals were minimised while the interior residual remained static. In many cases these led to sudden explosion of the loss value which corrupted the training. To avoid these, the ideal weight distribution was found to be $[1, 0.5, 0.5]$ for interior, boundary and initial residuals respectively.

Proxy normalisation

It is suggested that the problem formulation is normalised such that the inputs and outputs of the PINN do not have large discrepancies.[56] To achieve this, transformations were defined in the in_tf and out_tf attributes of the initialised DNN. The input to the network was transformed using a modified version of minmax normalisation, eq.(3.10)

$$X_{norm} = 2 \times ((X_{in} - X_{min})/(X_{max} - X_{min})) - 1 \tag{3.10}$$

; where X_{in} is the original input, X_{min} and X_{max} are the lower and upper boundaries of the domain and X_{norm} is the normalised input being fed into the network. This ensures that each feature takes a value between -1 and 1. On the other hand, the output of the network is initially passed through a Softplus activation function before being scaled by a factor equal to the maximum temperature observed in the FDM experiments. In short, the output transformation, eq.(3.11)

$$X_{out} = T_{max} \times Softplus(X_{NN}) \tag{3.11}$$

; where X_{out} is the final value and X_{NN} is the original output of the network, ensures that the prediction is always positive (no negative temperatures can exist in the Kelvin scale) and the network does not need to predict values larger than 1.

Thermal-dependent material properties

To incorporate the thermal-dependent material properties in the framework, an algorithm named *temp_var_pick* had to be implemented which runs in parallel with the training of the model. As soon as a prediction u is made, it is passed into the *temp_var_pick* function; to find the properties of the material at the predicted temperature, u is compared with the temperatures of the known material states and the closest candidate is identified using Numpy's $argmin$ function. The material properties associated with the selected candidate are then returned to the residual function for the necessary calculations.

It should be noted that 15 material states at equally spaced intervals between $T = 20K$ to $T = 2300K$ were initially given. The steps induced from these, averaged at 0.01113 J/gK and 0.00124 W/mmK for c_p and k respectively, were too big for smooth training. To improve, the known values were interpolated, Figure 3.9, such that a total of 150 states were input to the framework. This significantly improved the final result; The material properties at different temperatures should be given at the smallest possible temperature intervals, or even better, as a continuous function $Props = f(u)$.

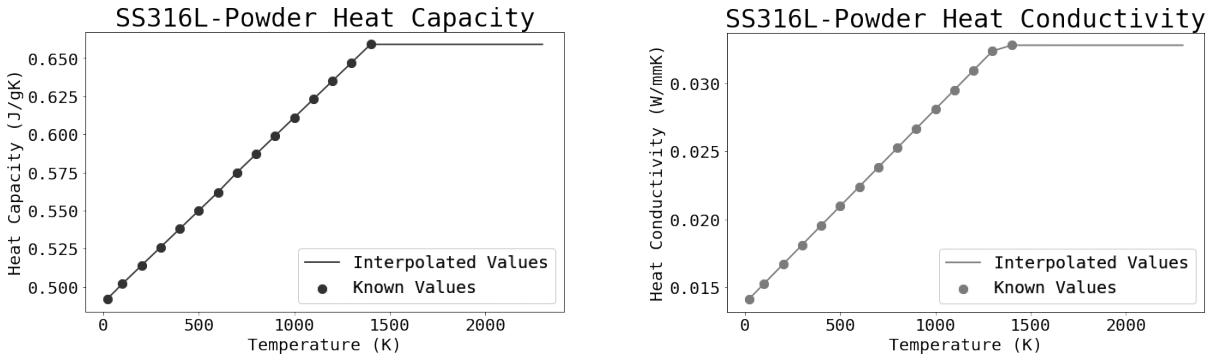


Figure 3.9: Thermal-dependent properties of SS316L-Powder

Main network architecture

The architecture of the main network is detrimental to the model's representation capacity, as it directly affects the amount of trainable parameters. For experiment PINN-ENA, even a small architecture with 3x(20) hidden layers lead to adequate training. However, it was also noted that smaller architectures were more prone to sub-optimal convergence. Sub-optimal convergence is also attributed to the nonconvexity of the problem, meaning that the network's initial weights are detrimental to the final approximation. To mitigate this, Xavier initialisation is used for all weights, and all bias values are adjusted to 0.

On the same note, better convergence from larger main networks came at a cost. For one, they trained much slower when compared to simpler versions, posing an issue given that training time is of essence. In addition, it was found that larger architectures exponentially increased the computational cost of the framework's training. This is directly correlated to the use of automatic differentiation and constitutes one of the main challenges in the next experiments, PINN-DYO and PINN-TRIA.

Activation functions

An appropriate activation function prevents linearity. In order to investigate the suitability of different activation functions for this problem, various tests were performed to examine their isolated effects. Figure 3.10 shows the evolution of training loss (total, co-

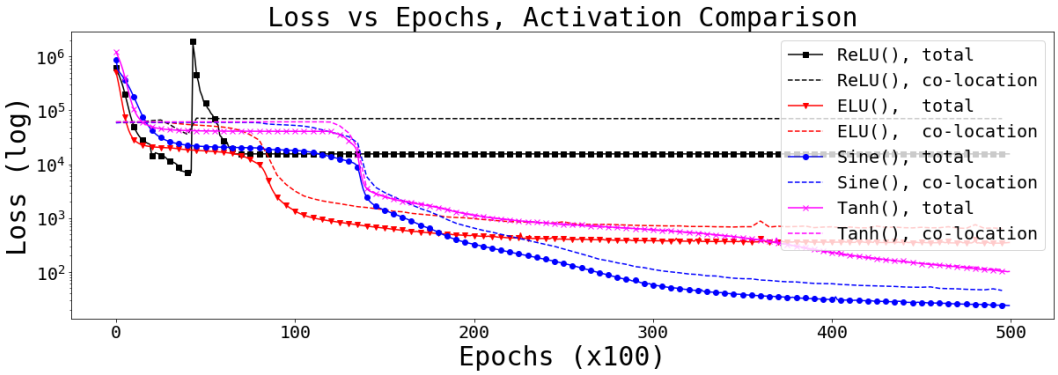


Figure 3.10: Loss evolution during training with sine, hyperbolic tangent, rectified linear unit and exponential linear unit activation functions.

location) across epochs with ReLU, ELU, Sine and Tanh activation functions. It is clear that Sine, ELU and Tanh exhibit better behaviour than ReLU. This highlights one of the peculiarities of PINN as the most common activation function is impractical in their case. Upon further research, it was found that this occurs because the second derivative of the ReLU function is equal to 0 which augments the vanishing gradient issue during the residual calculation (which includes multiple second derivative terms). ELU's second derivative is also equal to 0 for $x > 0$, hence the reduced performance after 20,000 epochs.

To compare the Tanh and Sine activation functions the experiments were repeated five times to ensure repeatability. In terms of training loss, Sine activation led to marginally higher performance in all tests, with final loss equal to 7.68 ± 1.32, compared to Tanh with with $5.10e+1 \pm 3.36$. This could be due to the periodic nature of the sinusoidal function which oscillates between -1 and 1 for $-\infty < x < \infty$, whereas the hyperbolic tangent function takes asymptotic values at relatively small values of x. In terms of training speed, the Tanh activation was always faster, at a rate of $50,000$ epochs in $2h14m \pm 3m$ in contrast to $2h32m \pm 2m$. The computational complexity for both functions is the same, hence the difference is attributed to their implementation; the former is native to the PyTorch library whereas the the latter is part of a 3rd-party library called SIREN.

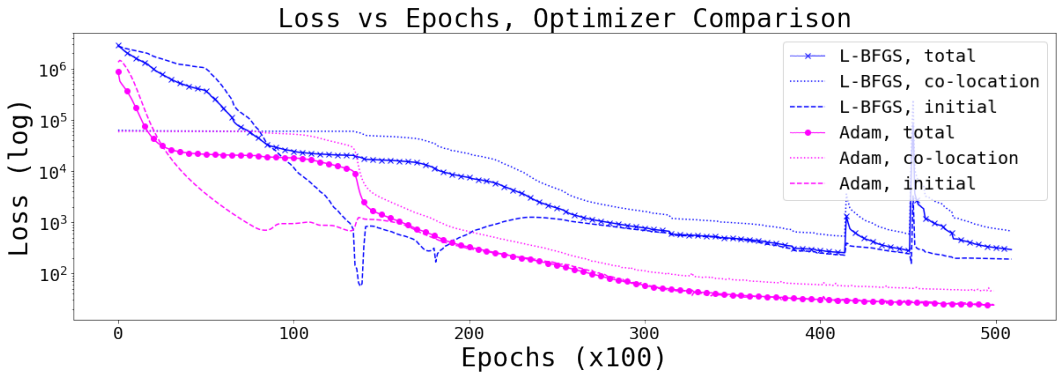


Figure 3.11: Loss evolution during training with Limited-BFGS and Adam optimizers.

Optimisation algorithm

The optimiser used for the minimisation of the loss function is also an important tool. In this section the Limited-BFGS (L-BFGS), Adam and Stochastic Gradient Descent (SGD) algorithms were compared. Implementing the L-BFGS optimiser required more steps, owing to the large number of parameters that needed to be tuned. Most importantly, the vanilla flavour of SGD failed to train the network in all attempted experiments, possibly due to its naivety in tackling the vanishing/exploding gradient problem.

Isolated results for L-BFGS and Adam can be seen in Figure 3.11. While both optimisers yielded satisfactory results in terms of final loss, the L-BFGS optimiser resulted in a rather noisy loss evolution, raising concerns regarding the settings used. No attempted change seemed to improve this behaviour. On the contrary, according to [57], L-BFGS is currently the most critical technology for PINN and a lot of online scientific communities tend to agree. Research revealed that PyTorch's implementation of the algorithm is not exactly correct and that double precision floats should be used for better results. However, double precision floats are memory intensive and the suggestion was rejected. To improve this, a TensorFlow implementation of the optimiser could be used. Finally, Adam not only performed better but also faster. This is attributed to the increased complexity of the L-BFGS optimiser, which is a quasi-netwon method that utilises historical data during training.

3.3.2 PINN-DYO: Hybrid Approach for High-Fidelity

The idea of augmenting the high-fidelity experiment from Section 3.2.3 into the continuous realm inspired experiment PINN-DYO. In 3.2.3, integrating the variations of the heat source significantly increased the system's complexity. Given the significant temporal dependencies and extra non-linearities, the same behaviour is predicted here.

Some simplifications were made. Firstly, the domain size was decreased from (10, 5, 5) mm to (5, 2, 2) mm. This did not affect the experiment as the extracted path still fit in the specified space. Secondly, the size of the secondary proxy distribution near the source was modified such that its shape and location dynamically changed, depending on the source settings at each given instance. Thirdly, the source calculation was changed from eq.(3.2) to eq.(3.4). Finally, the lambda weights of the loss function were re-tuned, such that even more attention is brought to the interior residual term.

After these changes, the network seemingly trained, Figure 3.12 ('no data'), but during inference this was far from right. Specifically, the network learned the track of the laser but the melt pool's peak temperatures were smoothed out. This was attributed to the high gradients required near the peak temperatures, which were not learned by the network because the interior proxy distribution remained sparse even after domain shrinkage. Here it shoud lbe noted that the interior proxy distribution had to change at a higher frequency to capture all laser locations, implying that for a constant number of proxies, fewer samples are associated with each time-step. To solve this, the amount of proxies was increased from 100.000 to 500.000 and a higher allocation rate was allowed for the secondary distribution. Major improvements were noticed, however, there was still gap for improvements.

Coming back to Karniadakis et al., the idea of a hybrid data-driven approach was reconsidered. This required the inclusion of a small amount of known ground truth data in the training procedure, embedded into the loss function as an additional 4^{th} term. A training data-set was created using the high-fidelity FDM simulation at a low resolution, such that

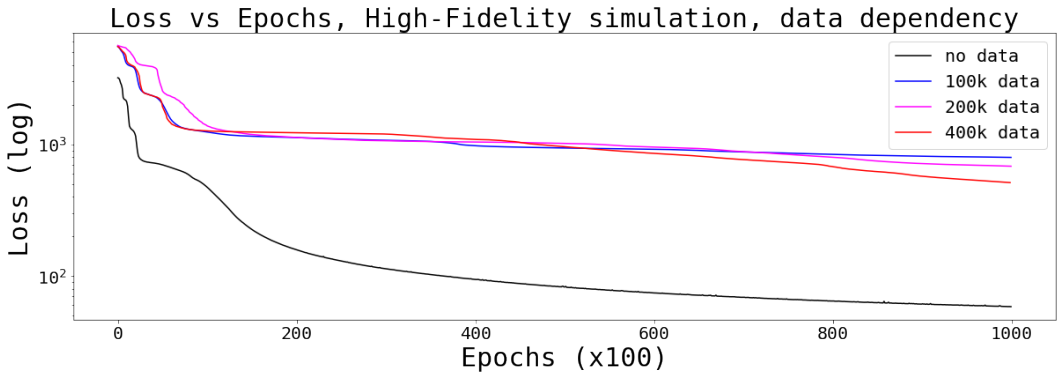


Figure 3.12: Loss evaluation of high-fidelity experiment for different data dependencies.

data collecting took less than 10 minutes. The data-set could be loaded directly on the CPU as it doesn't make use of automatic differentiation, avoiding additional workload from being added on the GPU. Multiple experiments with different amounts of training data were conducted, with results as in Figure 3.12.

During inference, for $n_{data} > 200.000$ the PINN solution started taking an acceptable shape; more on this follows in the Results section. A short video was prepared to visualise the PINN results of the high-fidelity experiment and is available at High-fidelity PINN.

3.3.3 PINN-TRIA: Parameterised Solver

After reproducing the single-track and high-fidelity deposition experiments using PINN, the fourth objective was targeted. Specifically, the question is whether a generalised solver is viable. A generalised, or parameterised, solver should be able to learn a family of solutions over a parameter space. In this scenario, the parameter space is defined by all conceivable combinations of process settings. The most important settings of the LPBF system are the laser's power, speed and focus. These settings are to be used in the NN as additional inputs and taken into account during the temperature prediction.

Here, it should be noted that HyperPINN have previously been suggested as a PINN flavour that can parameterise PDE.[58] However, constructing the necessary hyper-networks is difficult and necessitates the use of extra software that runs in the background during

Algorithm 1 Pipeline used for parameterised training of physics-informed neural network.

```
procedure PARAMETERISED PINN TRAINING
    get global proxies P
    set settings range S
    while iteration < I do                    ▷ I = number of training epochs
        cost = 0
        for setting-set in S do
            if memory limit not reached then
                calculate cost C_set for P with given setting-set
                cost += C_set
            else
                cost.backward()
                cost = 0
        cost.backward()
```

training and inference. As such, exploring other routes was preferred.

Initially one key setting was identified that would be most useful if parameterised. This was primarily done to learn more about the underlying principles of a parameterised PDE while also limiting the complexity of the problem until more experience was obtained. From feedback, it was concluded that the laser focus would be best, as existing experimental data could help validate the results at a later stage of this research. The focus is directly correlated with the laser's radius r in eq.(3.2).

Initial experiments involved the generation of proxies which included an additional feature with a random value between 0.1 and 0.4 such that the proxy structure took the form (x, y, z, t, f). This additional feature represented the physical value of the laser focus in millimeters, input into the network and used in the source calculation. These experiments were fruitless; due to the noise in the focus feature, the network was obliged to try mapping each proxy to a solution rather than directly optimising its gradients.

Using a trial-and-error technique, it was discovered that a large number of proxies with the same setting-set must be processed at once, to generate momentum leading to gradient-based training. However, more than one setting-set must be processed per training iteration, such that the network generalises beyond seen setting values. To visualise the

Table 3.5: Training range of parameterised settings.

	Lower	Upper
Power (W)	180	220
Velocity (mm/s)	200	600
Focus (mm)	0.2	0.4

proposed pipeline a pseudo-code has been prepared, Algorithm 1. This idea yielded very good results with acceptable loss convergence. The memory limit if-loop has been introduced such that the size of the cost graph does not exceed the memory capacity of the used GPU while also ensuring each iteration is as efficient as possible.

The trained model was able to output expected and logical temperature profiles for different focus values. Herein, the same idea was applied to parameterise the solution with respect to laser focus, power and speed simultaneously. The settings range was strategically selected such that real-use values were included, Table 3.5. Some other modifications were also required; most importantly, three additional input nodes were added on the NN; the structure of the input takes the form (x, y, z, t, f, P, v) In addition, the secondary distribution of the co-locations had to dynamically change position along the x-axis, depending on the input velocity. This was necessary such that the second distribution remains on the laser's track at all assumed speeds to capture the high gradients. Finally, the size of the domain was reduced to (6,1.5,1) mm.

The training took longer than expected, lasting more than one and a half days for 40,000 epochs. This was due to the fine discretisation of the parameter space, which resulted in over 200 different setting combinations that had to be investigated at least once during each iteration. The silver lining is that during training the loss started reducing from 6.13e+7 until it converged at 1.61e+2. Although not directly comparable with the results obtained from PINN-ENA, the large reduction is promising.

Chapter 4

Results and Discussion

In this section the PINN models, summary in Table 4.1, are evaluated. The evolution of the training loss for each PINN is also plotted, Figure 4.1. Any fluctuations are attributed to the parallel optimisation of the three loss terms which may cause temporal disentanglement from local minima reached in previous steps.

In the following subsections, the FDM results are used as ground truth to assess the accuracy of the different surrogate models presented in Section 3.3. The learning transfer capabilities of PINN are also put into test to re-train the PINN for different in-process parameter values in a fraction of the required time. Finally, the generalised solver is compared with multiple FDM versions of the single-track deposition experiment before using it to extract melt pool features.

Table 4.1: Final PINN-ENA, -DYO, -TRIA configurations used during testing.

Parameter	PINN-ENA	PINN-DYO	PINN-TRIA
Architecture	4 + (3 x 20) + 1	4 + (6 x 20) + 1	7 + (6 x 20) + 1
Learning rate	5e-5	2e-5	1e-5
Epochs	50.000	120.000	40.000
Optimiser	Adam	Adam	Adam
Activation	Sinusoidal	Sinusoidal	Sinusoidal

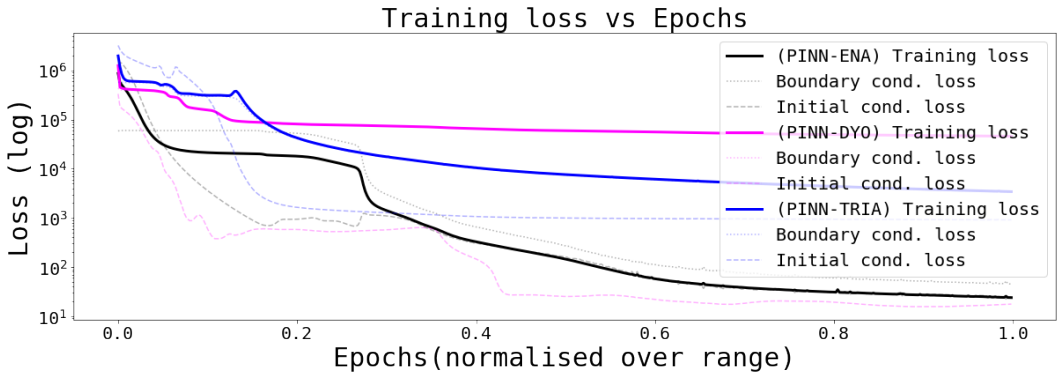


Figure 4.1: Final configurations of physics-informed neural networks used for reproduction of conducted experiments.

4.1 Benchmark Evaluation

Single-track deposition experiments

Figure 4.2 compares thermal profiles obtained from FDM and PINN-ENA at settings ref.#1 and times $t = 0.013, 0.052$ and 0.093 seconds. It can be seen that the PINN has generalised well with reasonable outputs at arbitrary times. Small differences between the 2 solutions are expected as the surrogate model training does not converge with zero loss. This gap has been quantified as equal to 1.48%, using the relative mean absolute error (MAE) calculation, eq.(4.1)

$$d_{MAE} = \frac{1}{N} \times \sum_{i=0}^{N} \frac{|T_{PINN} - T_{FDM}|}{|T_{FDM}|} \times 100 \tag{4.1}$$

; where T_{PINN} and T_{FDM} are the temperature predictions obtained from the two solvers at all N locations for which an FDM solution exists. These N locations (x, y, z, t) can be considered previously unseen from the NN because the co-location points were arbitrarily generated. Given this, and with the maximum local error being equal to 21.23 K (0.013% from highest temperature) the solution gap can be considered negligible.

It can also be seen that the wake of the melt pool in all PINN thermal profiles is not symmetric along the x axis. This is the major error source and it should be noted that it

does not directly affect the melt pool profile.

At time $t = 0.013$ seconds a large difference is reported near the x=0 boundary. This is in fact one of the advantages of PINN which do not suffer from hard-enforced boundary conditions. The FDM solutions near the physical boundaries can't be considered accurate, as the temperature values u at $x = 0$ are directly cloned from the values at $x = 0 + dt$ such that $\partial u/\partial x = 0$. This also applies for all other sides of the domain.

Finally, PINN-ENA was tested on a new set of proxies located at arbitrary (x, y, z) locations and time $t = 0$. This was done to examine how well the network generalised for the initial conditions; the MAE score for the initial condition test data-set was 0.23% (i.e., 99.76% accurate) and can be considered excellent. This brought the PINN-ENA assessment to a close, emphasising the proposed framework's exceptional capability in mimicking single-track deposition simulations.

High-fidelity experiments

A similar test routine was conducted for PINN-DYO. In Figure 4.3 the FDM solution is compared with the surrogate model outputs at times $t = 0.0010$, 0.0038, 0.0051, 0.0073 and 0.0097 seconds. It can be seen that as time passes the two solutions diverge. This is particularly visible at the wake of the melt pool. Specifically, during the first half of the simulation ($t < 0.005$) the MAE score of the PINN solution is 3.5% while for the second half the MAE score spikes at 10.6%. Errors like this were expected, as the training of PINN-DYO finished with relatively high losses driven by high interior residuals, however, the reason for this time dependency remains unclear. Despite large MAE loss, it should be noted that the maximum local error remained relatively low at 24.51 K and the network predicted initial temperatures on unseen locations with 97.33% accuracy.

Improvements could be made such that the final solution becomes more accurate, by training for more epochs, training a larger network or by including more ground truth data. It is argued that further optimising this experiment was beyond the scope of this

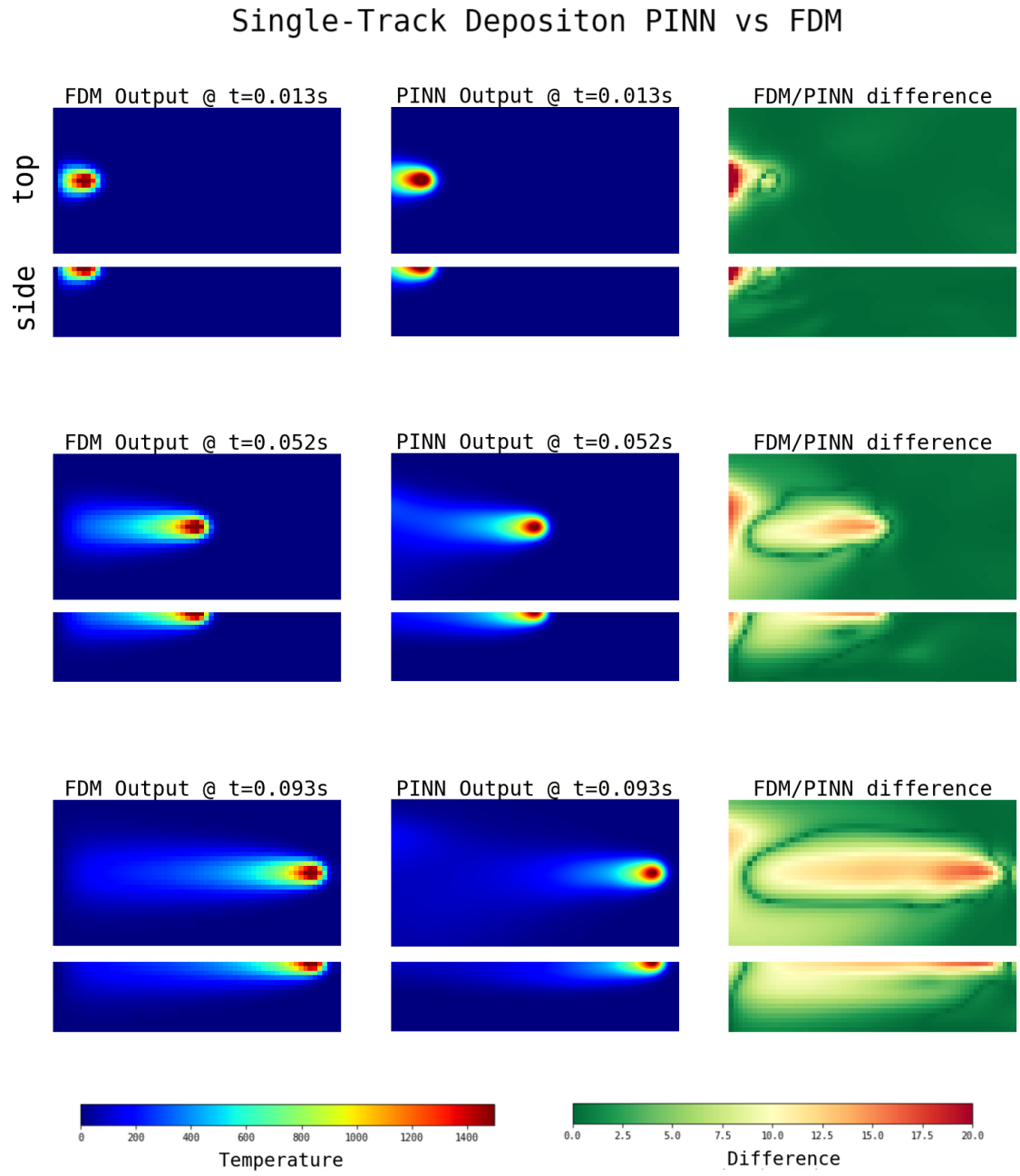


Figure 4.2: Single-track deposition thermal profiles obtained from the numerical method and the surrogate model (PINN-ENA) at settings ref.1.

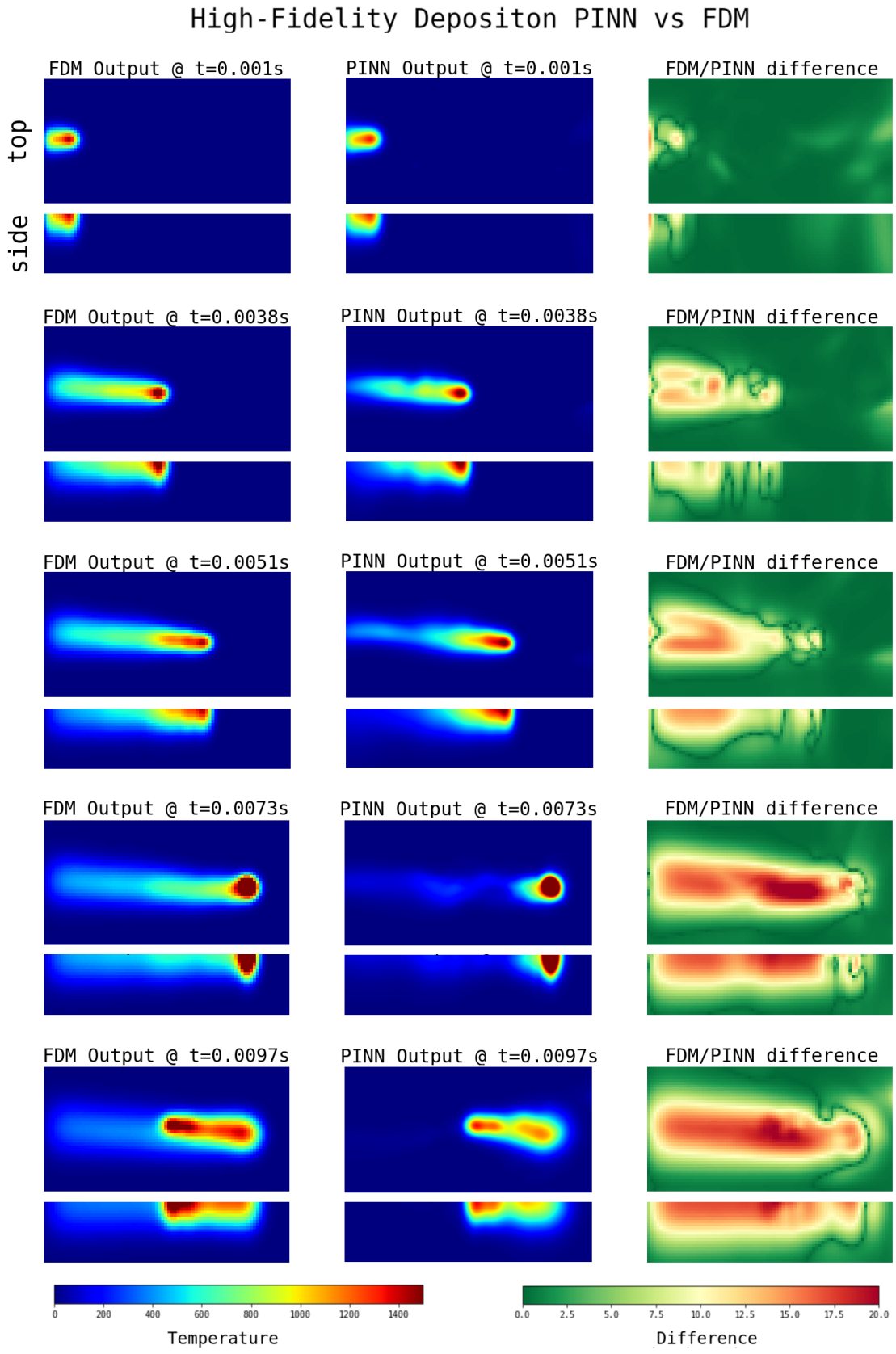


Figure 4.3: High-Fidelity deposition thermal modeling results obtained from the numerical method and the surrogate model (PINN-DYO).

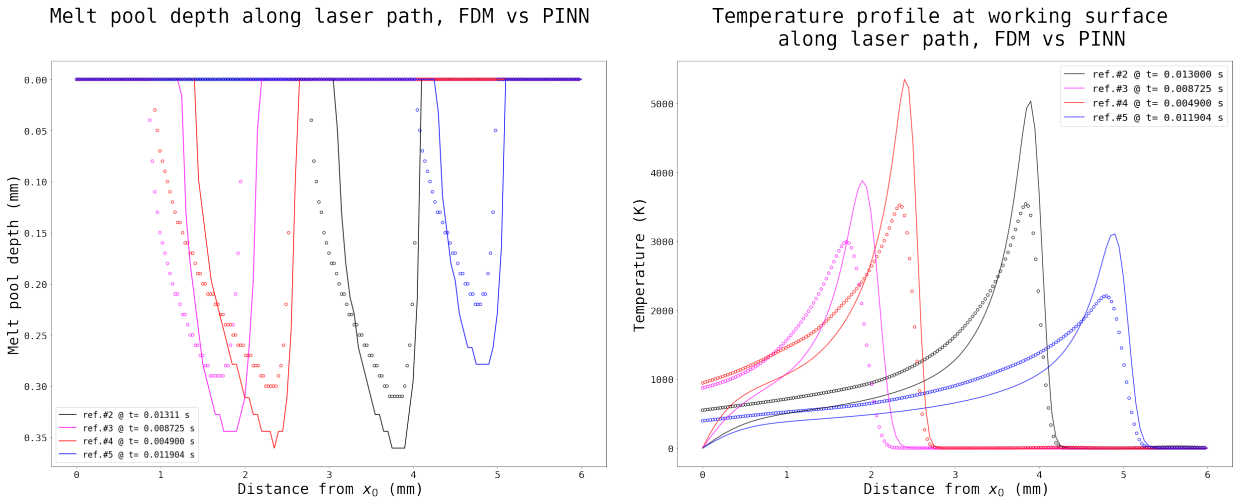


Figure 4.4: Comparison of depth and temperature profiles along laser track for referenced experiment parameters using finite difference method simulations and a generalised physics-informed solver.

research, mainly due to the combination of high training times (in the magnitudes of 5 to 6 hours for 100,000 epochs), the increased GPU workload from 500,000 proxies and the inaccuracies discussed above. However, this is the first time a PINN has been used to replicate a dynamic high-fidelity thermal simulation and would be interesting to examine how it may be employed in situations where training time is not a problem or a high resolution solution is mandatory.

Parameterised Solver

PINN-TRIA outputs were compared with the solutions obtained from the FDM simulations at all referenced settings, Figure 4.4. It should be emphasised that all of the PINN predictions in the figure were generated in less than 10 seconds in contrast to the FDM solutions which took more than 10 hours, as mentioned in Section 3.2.2.

Although the generalised solution is not fully precise, the observed patterns are highly promising. For one, the surrogate model generalised well in terms of predicting the laser's position. This is justified from the fact that for all experiment settings and for all arbitrary times the surrogate model generates outputs with high gradients at the expected locations.

For two, there is a strong correlation between the maximum depth magnitudes at all settings. Specifically, maximum depth is calculated with up to 86% accuracy.

In terms of temperature accuracy, peak values are observed at the expected locations but once again, the actual values are smoothed out. The disparities indicate that there is room for improvement in the PINN, but the hard-encoded boundaries used in FDM raise the question of how the evolution of the theoretical solution is influenced. The same improvements suggested for PINN-DYO can be applied here. It is also suggested that a verified and more sophisticated numerical analysis is made available for validation, such that uncertainties like this are avoided. On a positive note,it can be seen that the generalised solver does not suffer from hard-enforced boundary conditions at $x = 0$ mm.

4.2 Meta-learning

In this section the meta-learning, or learning transfer, capabilities of PINN are investigated and discussed. The motivation behind these tests is to fine-tune a PINN-ENA parent model on different process parameters in a fraction of the time it took to train it in the first place. To do this, the parent model is loaded and re-trained using new values for laser power, speed and material. To change the material, Ti64 (titanium) thermal-dependent properties were found online and processed as explained in 3.3.1.(Thermal-dependent material properties) before being fed into the framework.

Figure 4.5 compares the training loss evolution of six offspring networks to that of their parent. Note that the epoch axis for the parent is scaled by 100, whereas the epoch axis for the children is scaled by 10.

Child1 trains in less than 1000 epochs ($\approx$1 minute) for a different value of laser power. When compared to the parent model, Child2 and Child3 train rather quickly; however, when compared to Child1, they train at a slower rate since both laser power and speed are modified concurrently. Ideally, only one parameter should be modified during fine-tuning;

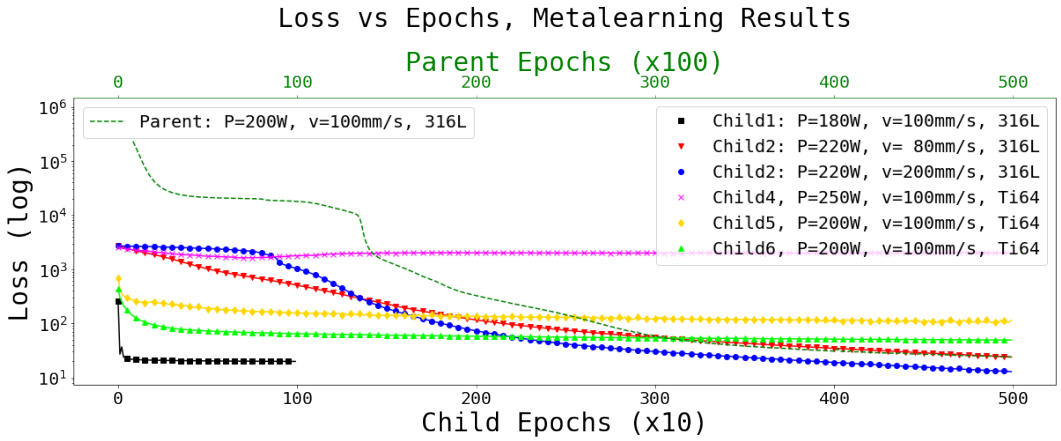


Figure 4.5: Metalearning loss curves of parent and children (re-trained) models.

to adjust two parameters a two-stage meta-learning procedure is suggested. Interesting would be to examine the viability of a comprehensive library of trained models for a wide range of parameters, that could be generated using a smart graph-search technique.

For Child4, 5 and 6 the material of the framework was switched from SS316L to Ti64. Child4 completely fails to re-train and Child5 does not converge well. This indicates that changing the material of the problem is not as easy as changing power or velocity, probably because the material characteristics contain non-linearities. Child6, however, converges well - in fact, Child6 is the byproduct of a second larger parent network with 6x(40) hidden layers. Larger architectures exhibit more malleability during meta-learning; when large power, speed and material offsets are needed during fine-tuning, a larger network is expected to yield better results.

4.3 Melt Pool Exploration

The author suggests that the generalised solver (PINN-TRIA) presented in this report is a powerful tool. Scan speed, laser power, and focus, all have a substantial influence on the process, which can now be assessed immediately. This section will offer a few examples of how this might be beneficial. For context, in Figure 4.6 some thermal profiles are visualised based on the referenced experiment settings used in Section 3.2.2. These

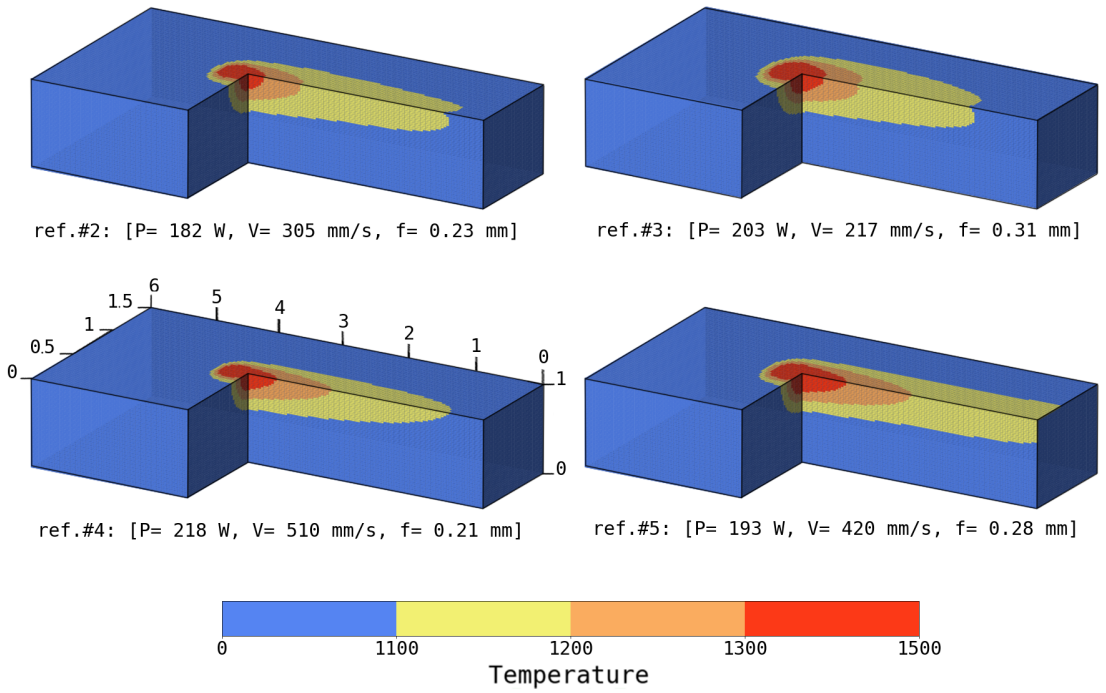


Figure 4.6: Melt pool visualisations for referenced experiment parameters retrieved instantaneously from a generalised physics-informed solver.

highlight the capability of the parameterised solver to offer three dimensional melt pool structure based on the three input parameters (P, v, f).

For a given combination of process settings, an algorithm has been scripted to generate a thermal profile prediction for the whole domain using the generalised solver. The time of the prediction is a function of the input velocity, such that the melt pool is roughly positioned at (4,0.75,1) mm. This guarantees that the melt pool remains steady during the analysis, allowing for more effective interaction with it. The generated 3D thermal profile is then filtered using Numpy's *where* function, such that regions with temperatures above the melting point of SS316L (= 1375 K) are activated (= 1) and all other regions are deactivated (= 0). This produces a cluster which represents the melt pool. The required features (melt pool length, width, depth at any location and volume) can then be extracted using a series of Numpy functions and grid manipulations.

To examine how the melt pool changes with the isolated variation of the each laser setting, Figure 4.7 has been prepared. In general, it can be seen that a lower focus value, a lower

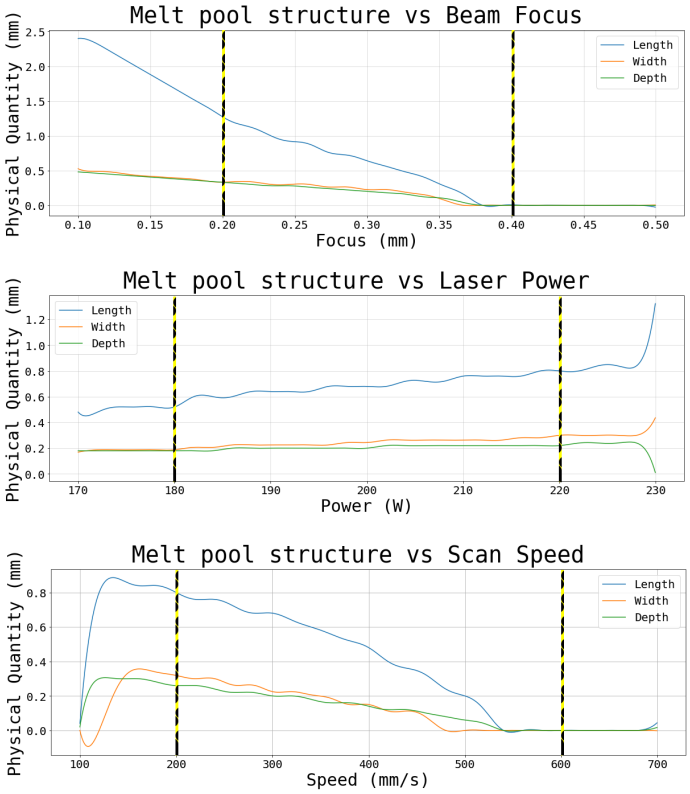


Figure 4.7: Melt pool structure variations driven by individual parameter value scans while the rest remain constant ($P = 200W, v = 300mm/s, f = 0.3mm$).

velocity and a higher power increase the size of the melt pool in all considerable metrics. This is expected behaviour; i) as the focus value decreases the power of the laser is concentrated in a smaller volume resulting in the melting of more material, ii) as the scanning speed decreases, the energy input rate at the location increases resulting in the melting of more material and iii) as the power of the laser beam increases, more energy is directly delivered on the working substrate resulting in the melting of more material. The observed patterns seem to obey known physics, however, experimental validation is still required to confirm how well they represent real-life phenomena.

It is also observed that the trends only hold for a tiny margin beyond what has been taught. This indicates that the training parameter boundaries must be strategically selected prior training. The range between lower and upper boundaries is also expected to affect convergence. For instance, it is suggested that using a lower range for scanning speed could improve the results presented above.

Chapter 5

Conclusion

5.1 Summary of Thesis Achievements

The implementation of a basic physics-informed neural network in the context of AM allowed for the examination of its behaviour under various design scenarios. It is also the first time a high-fidelity deposition experiment is replicated in continuous space-time, utilising data obtained from a co-axial imaging system. Although the solution diverges for times greater than 0.005 seconds, enhancements are suggested for scenarios where training time is not critical. Most importantly, the first parameterized PINN in the context of AM is presented in detail. It can instantly predict thermal profiles and melt pool characteristics (such as length, width, depth, volume) in practically zero computational cost. Its outputs show good resemblance with theoretical ground-truth obtained from numerical solvers. Such tool can be used in a multitude of ways; for depth mapping, physics-intuitive feature extraction, thermal gradient prediction, etc. Following experimental validation, it may potentially be utilised directly in a feed-forward control loop to modify process parameters such that the melt pool maintains its stability. PINN just recently gained prominence; the provided analysis should be useful to other AM engineers who lack AI-engineering competence but wish to put them to the test.

5.2 Limitations and Future Work

There is margin for improvements to be made. Firstly, the L-BFGS algorithm could help for better convergence. In addition, other heat source models could be used to examine their behaviour. To do that, the implemented code allows modifications for other volumetric heat sources to be seamlessly incorporated. The results of the parameterised PINN might potentially be experimentally tested in future studies. Alternatively, the parameterised PINN outputs could be compared with solutions of verified numerical simulations.

Using automatic differentiation was a challenge. Although the implementation of the function is straight-forward, the calculation of partial derivatives is a computational burden. The used setup (GPU: Nvidia GeForce RTX 2080 Ti (12 Gb), CPU: Intel Xeon W-2123 (8-core @ 3.60 GHz), RAM: 62.5 Gb) occasionally failed to devote adequate memory, especially as the amount of training proxies or the size of the network increased. It is suggested that data parallelism is employed to train more sophisticated networks with larger capacity, allowing for the inclusion of additional physical phenomena in the analysis. Data parallelism refers to the process of collaboratively training the same network on multiple GPUs, each loaded with a subset of the available data.

Finally, it should be noted that the applications of PINN in LPBF are not limited to thermal analyses. PINN are powerful networks that can also solve fluid problems; similar to what Cai et al. did in [59], a PINN could be used to model the plume formation and flow. This might shed light on the plume-laser interaction; for example, data-driven parameter discovery can help quantify laser attenuation from different amounts of plume. Such analysis could be performed with standard cameras, but a Background Oriented Schlieren Imaging system can kick-start the idea. This particular imaging system is suggested as it is capable of detecting changes in the refractive index of air (or inert gas in the case of AM), making it extremely capable in visualising hot smoke.

5.3 Ethics and Legal Considerations

As mentioned in Chapter 1, AM is rapidly gaining attention and for good reason; it gives its users the ability to manufacture virtually anything. This power comes with immense responsibility, and as technology advances, scientists must ensure that the infrastructure around it is bullet-proof. The work given in this study should not be used to augment AM for immoral reasons such as the unauthorised manufacture of firearms, biological organs, or any other products that are part of intellectual property.

The utilisation of high-powered laser systems was part of the safety concerns, however, no interaction with such systems was required during the experiments for this study. All available data were made available from the Mechanical Engineering Department of Imperial College London.

Finally, the AI procedures described in this study do not include any human individuals or their personal information. As such, the AI aspect of this work is ethically and legally neutral.

Bibliography

[1] V. R. Shulunov, "Several advantages of the ultra high-precision additive manufacturing technology," *The International Journal of Advanced Manufacturing Technology*, vol. 85, no. 9-12, pp. 1941–1945, 2015.

[2] N. P. Calta, J. Wang, A. M. Kiss, A. A. Martin, P. J. Depond, G. M. Guss, V. Thampy, A. Y. Fong, J. N. Weker, K. H. Stone, C. J. Tassone, M. J. Kramer, M. F. Toney, A. Van Buuren, and M. J. Matthews, "An instrument for in situ time-resolved x-ray imaging and diffraction of laser powder bed fusion additive manufacturing processes," *Review of Scientific Instruments*, vol. 89, no. 5, p. 055101, 2018.

[3] J. Yang, L. Chen, W. Huang, and Y. Li, "Survey on artificial intelligence for additive manufacturing," pp. 1–6, 09 2017.

[4] A. Gajakosh, R. Kumar, M. Vinayagam, R. Shanmugam, and M. Ramoni, *Applications of Artificial Intelligence in Additive Manufacturing*. 12 2021.

[5] D. A. J. Brion and S. W. Pattinson, "Generalisable 3d printing error detection and correction via multi-head neural networks," *Nature Communications*, vol. 13, p. 4654, Aug 2022.

[6] S. Larsen and P. A. Hooper, "Deep semi-supervised learning of dynamics for anomaly detection in laser powder bed fusion," *Journal of Intelligent Manufacturing*, vol. 33, pp. 457–471, Feb 2022.

[7] W. Wang and S. Y. Liang, "A 3d analytical modeling method for keyhole porosity prediction in laser powder bed fusion," *The International Journal of Advanced Manufacturing Technology*, vol. 120, pp. 3017–3025, May 2022.

[8] A. Gaikwad, R. Yavari, M. H. Montazeri, K. D. Cole, L. Bian, and P. K. Rao, "Toward the digital twin of additive manufacturing: Integrating thermal simulations, sensing, and analytics to detect process faults," *IISE Transactions*, vol. 52, pp. 1204 – 1217, 2020.

[9] M. Tang, P. Pistorius, and J. Beuth, "Prediction of lack-of-fusion porosity for powder bed fusion," *Additive Manufacturing*, vol. 14, 01 2017.

[10] P. Honarmandi, R. Seede, L. Xue, D. Shoukr, P. Morcos, B. Zhang, C. Zhang, A. Elwany, I. Karaman, and R. Arroyave, "A rigorous test and improvement of the eagar-tsai model for melt pool characteristics in laser powder bed fusion additive manufacturing," *Additive Manufacturing*, vol. 47, p. 102300, 2021.

[11] K. L. Ness, A. Paul, L. Sun, and Z. Zhang, "Towards a generic physics-based machine learning model for geometry invariant thermal history prediction in additive manufacturing," *Journal of Materials Processing Technology*, vol. 302, p. 117472, 2022.

[12] Y. Du, T. Mukherjee, and T. DebRoy, "Physics-informed machine learning and mechanistic modeling of additive manufacturing to reduce defects," *Applied Materials Today*, vol. 24, p. 101123, 2021.

[13] Q. Zhu, Z. Liu, and J. Yan, "Machine learning for metal additive manufacturing: predicting temperature and melt pool fluid dynamics using physics-informed neural networks," *Computational Mechanics*, vol. 67, pp. 619–635, Feb 2021.

[14] D. Knüttel, S. Baraldo, A. Valente, K. Wegener, and E. Carpanzano, "Model based learning for efficient modelling of heat transfer dynamics," *Procedia CIRP*, vol. 102, pp. 252–257, 2021. 18th CIRP Conference on Modeling of Machining Operations (CMMO), Ljubljana, Slovenia, June 15-17, 2021.

[15] S. Liao, T. Xue, J. Jeong, S. Webster, K. Ehmann, and J. Cao, "Hybrid full-field thermal characterization of additive manufacturing processes using physics-informed neural networks with data," 2022.

[16] A. M. TURING, "I.—COMPUTING MACHINERY AND INTELLIGENCE," *Mind*, vol. LIX, pp. 433–460, 10 1950.

[17] N. J. Nilsson, "The quest for artificial intelligence: A history of ideas and achievements," 2009.

[18] Y. Wang, "Towards the synergy of cognitive informatics, neural informatics, brain informatics, and cognitive computing," *IJCINI*, vol. 5, pp. 75–93, 01 2011.

[19] I. M. Cockburn, R. Henderson, and S. Stern, "The impact of artificial intelligence on innovation," *National Bureau of Economic Research Working Paper Series*, vol. No. 24449, March 2018 2018.

[20] J. Jiménez-Luna, F. Grisoni, and G. Schneider, "Drug discovery with explainable artificial intelligence," *Nature Machine Intelligence*, vol. 2, pp. 573–584, Oct 2020.

[21] T. P. Nagarhalli, S. Mhatre, S. Patil, and P. Patil, "The review of natural language processing applications with emphasis on machine learning implementations," in *2022 International Conference on Electronics and Renewable Systems (ICEARS)*, pp. 1353–1358, 2022.

[22] A. J. Davison, "Mobile robot navigation using active vision," 1998.

[23] H.-Y. Huang, M. Broughton, M. Mohseni, R. Babbush, S. Boixo, H. Neven, and J. Mcclean, "Power of data in quantum machine learning," *Nature Communications*, vol. 12, 05 2021.

[24] D. Gunning, E. Vorm, J. Y. Wang, and M. Turek, "Darpa's explainable ai (xai) program: A retrospective," *Applied AI Letters*, vol. 2, no. 4, p. e61, 2021.

[25] G. Cybenko, "Approximation by superpositions of a sigmoidal function," *Mathematics of Control, Signals, and Systems (MCSS)*, vol. 2, pp. 303–314, Dec. 1989.

[26] I. J. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016. http://www.deeplearningbook.org.

[27] I. Lagaris, A. Likas, and D. Fotiadis, "Artificial neural networks for solving ordinary and partial differential equations," *IEEE Transactions on Neural Networks*, vol. 9, no. 5, pp. 987–1000, 1998.

[28] M. Raissi, P. Perdikaris, and G. Karniadakis, "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations," *Journal of Computational Physics*, vol. 378, pp. 686–707, 2019.

[29] J. Yu, L. Lu, X. Meng, and G. E. Karniadakis, "Gradient-enhanced physics-informed neural networks for forward and inverse PDE problems," *Computer Methods in Applied Mechanics and Engineering*, vol. 393, p. 114823, apr 2022.

[30] G. E. Karniadakis, I. G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, and L. Yang, "Physics-informed machine learning," *Nature Reviews Physics*, vol. 3, pp. 422–440, Jun 2021.

[31] E. Kroll and E. Buchris, "Weight reduction of 3d-printed cylindrical and toroidal pressure vessels through shape modification," *Procedia Manufacturing*, vol. 21, pp. 133–140, 01 2018.

[32] N. H. Cohrs, A. Petrou, M. Loepfe, M. Yliruka, C. M. Schumacher, A. X. Kohll, C. T. Starck, M. S. Daners, M. Meboldt, V. Falk, and W. J. Stark, "A soft total artificial heart—first concept evaluation on a hybrid mock circulation," *Artificial Organs*, vol. 41, p. 948–958, 2017.

[33] A. Haleem, M. Javaid, R. P. Singh, and R. Suman, "Significant roles of 4d printing using smart materials in the field of manufacturing," *Advanced Industrial and Engineering Polymer Research*, vol. 4, no. 4, pp. 301–311, 2021. 3D Printing of Polymers.

[34] D. K. Jayashankar, S. S. Gupta, L. Y. N. Stella, and K. Tracy, "3d printing of compliant passively actuated 4d structures," 2019.

[35] M. Vaezi, H. Seitz, and S. Yang, "A review on 3d micro-additive manufacturing technologies," *The International Journal of Advanced Manufacturing Technology*, vol. 67, 07 2012.

[36] A. Moetazedian, J. Allum, A. Gleadall, E. Mele, and V. V. Silberschmidt, "Matrex am: A new hybrid additive manufacturing process to selectively control mechanical properties," *Additive Manufacturing*, vol. 47, p. 102337, 2021.

[37] E. MacDonald and R. Wicker, "Multiprocess 3d printing for increasing component functionality," *Science*, vol. 353, no. 6307, p. aaf2093, 2016.

[38] D. Insights, "3d printing growth accelerates again," *TMT Predictions 2019*, 12 2019.

[39] R. McGuan, P. Kavehpour, and R. N. Candler, "Precision enhancement of 3d printing via in situ metrology," 2018.

[40] C. Margadji and J. M. Herreros, "Characterisation of nanoparticle emissions released from fused deposition modeling," 2021.

[41] A. Sola and A. Nouri, "Microstructural porosity in additive manufacturing: The formation and detection of pores in metal parts fabricated by powder bed fusion," *Journal of Advanced Manufacturing and Processing*, vol. 1, no. 3, p. e10021, 2019.

[42] S. Cao, "Review of laser powder bed fusion (lpbf) fabricated ti-6al-4v: process, post-process treatment, microstructure, and property," 2021.

[43] J. Yin, W. Zhang, L. Ke, H. Wei, D. Wang, L. Yang, H. Zhu, P. Dong, G. Wang, and X. Zeng, "Vaporization of alloying elements and explosion behavior during laser powder bed fusion of cu–10zn alloy," *International Journal of Machine Tools and Manufacture*, vol. 161, p. 103686, 2021.

[44] J. Liu and P. Wen, "Metal vaporization and its influence during laser powder bed fusion process," *Materials Design*, vol. 215, p. 110505, 2022.

[45] A. A. Martin, N. P. Calta, S. A. Khairallah, J. Wang, P. J. Depond, A. Y. Fong, V. Thampy, G. M. Guss, A. M. Kiss, K. H. Stone, C. J. Tassone, J. Nelson Weker, M. F. Toney, T. van Buuren, and M. J. Matthews, "Dynamics of pore formation during laser powder bed fusion additive manufacturing," *Nature Communications*, vol. 10, p. 1987, Apr 2019.

[46] E. Glaessgen and D. Stargel, "The digital twin paradigm for future nasa and u.s. air force vehicles," 04 2012.

[47] T. DebRoy, W. Zhang, J. Turner, and S. Babu, "Building digital twins of 3d printing machines," *Scripta Materialia*, vol. 135, pp. 119–124, 2017.

[48] B. Soundararajan, D. Sofia, D. Barletta, and M. Poletto, "Review on modeling techniques for powder bed fusion processes based on physical principles," *Additive Manufacturing*, vol. 47, p. 102336, 2021.

[49] P. Hooper, "Melt pool temperature and cooling rates in laser powder bed fusion," *Additive Manufacturing*, vol. 22, 05 2018.

[50] Z. Smoqi, A. Gaikwad, B. Bevans, M. H. Kobir, J. Craig, A. Abul-Haj, A. Peralta, and P. Rao, "Monitoring and prediction of porosity in laser powder bed fusion using physics-informed meltpool signatures and machine learning," *Journal of Materials Processing Technology*, vol. 304, 6 2022.

[51] E. Mirkoohi, D. E. Seivers, H. Garmestani, and S. Y. Liang, "Heat source modeling in selective laser melting," *Materials*, vol. 12, no. 13, 2019.

[52] Z. Zhang, Y. Huang, A. Rani Kasinathan, S. Imani Shahabad, U. Ali, Y. Mahmoodkhani, and E. Toyserkani, "3-dimensional heat transfer modeling for laser powder-bed fusion additive manufacturing with volumetric heat sources based on varied thermal conductivity and absorptivity," *Optics Laser Technology*, vol. 109, pp. 297–312, 2019.

[53] J. A. Goldak, A. P. Chakravarti, and M. Bibby, "A new finite element model for welding heat sources," *Metallurgical Transactions B*, vol. 15, pp. 299–305, 1984.

[54] S. Shrestha and K. Chou, "A study of transient and steady-state regions from single-track deposition in laser powder bed fusion," *Journal of Manufacturing Processes*, vol. 61, pp. 226–235, 2021.

[55] C. Tenbrock, T. Kelliger, N. Praetzsch, M. Ronge, L. Jauer, and J. H. Schleifenbaum, "Effect of laser-plume interaction on part quality in multi-scanner laser powder bed fusion," *Additive Manufacturing*, vol. 38, p. 101810, 2021.

[56] S. Cai, Z. Wang, S. Wang, P. Perdikaris, and G. Karniadakis, "Physics-informed neural networks (pinns) for heat transfer problems," *Journal of Heat Transfer*, vol. 143, 03 2021.

[57] S. Markidis, "The old and the new: Can Physics-Informed Deep-Learning replace traditional linear solvers?," *Front Big Data*, vol. 4, p. 669097, Nov. 2021.

[58] F. d. A. Belbute-Peres, Y.-f. Chen, and F. Sha, "Hyperpinn: Learning parameterized differential equations with physics-informed hypernetworks," 2021.

[59] S. Cai, Z. Wang, F. Fuest, Y. J. Jeon, C. Gray, and G. E. Karniadakis, "Flow over an espresso cup: inferring 3-d velocity and pressure fields from tomographic background oriented schlieren via physics-informed neural networks," *Journal of Fluid Mechanics*, vol. 915, p. A102, 2021.

Appendices

Appendix A: Porosity Classification

Data preparation

In the first phase of this project, an attempt to classify the optical data stream in 'good' and 'bad' frames was made. This binary classification aimed to reflect the probability of each pictured melt pool having caused a defect or not. One of the main challenges was to correlate the data collected from the in-situ monitoring system presented in Section 3.1, and a CT scan of the printed part, such that a labeled data occurs. However, no standard practices exist and a custom pipeline had to be prepared. The implemented pipeline consisted of 3 main steps; 1) registration of the two datasets, 2) removal of outliers and balancing of the data-set and 3) correlate individual pores with individual frames. More on each step follow below.

After labeling all melt pool frames, a NN was prepared and trained to find the hidden patterns. High accuracy in the magnitude of 80% was reported on a new unseen data-set, with a sample confusion matrix as seen in table 1. Despite good results, the localisation of the defects caused from the 'bad' frames remained a non-trivial task. The amount of steps needed to tackle the problem started increasing significantly and concerns regarding p-hacking were raised. As such, the focus of the research changed such that a self or semi-supervised approach is developed.

The porosity classifier is not discussed here as it is beyond the scope of this Appendix. It should, however, be noted that a PINI model with custom loss function was implemented, similar to McGowan E. et al., with negative impact on the accuracy. It is argued that this kind of approach

Table 1: Confusion matrix from porosity classification using labeled data obtained from the 3-stage pipeline (CT/optical data registration, spatial-proximity filter, turning point detector).

	P	N
T	479	675
F	100	236

is impractical from a mathematical point of view.

1. CT/optical data automatic registration

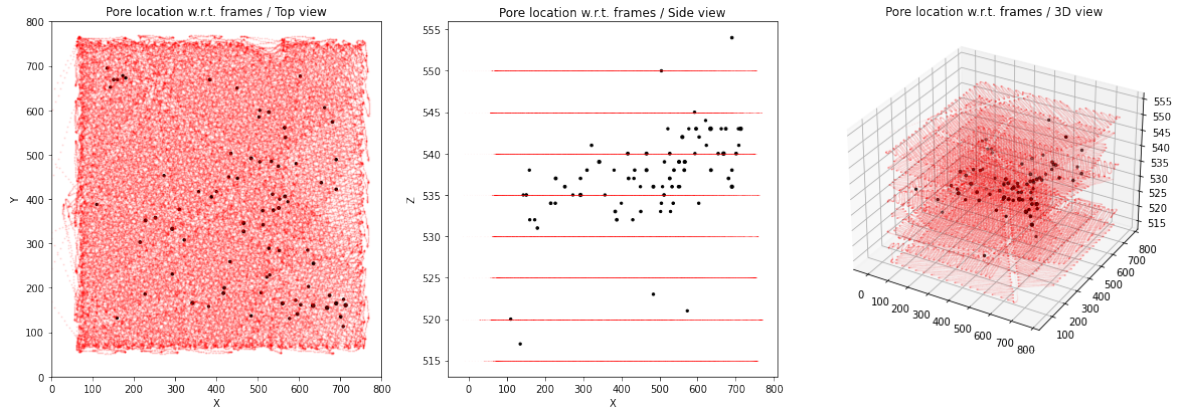
As the two data-sets come from different measuring devices, they do not share the same frame of reference. In order to relate them, one of the had to be scaled, translated and rotated so that specific coordinates of data-set A are directly related to the same coordinates of data-set B. The procedure to find this transformation is called data registration.

Probreg is probablistic point cloud registration library for Python, offering multiple registration solutions such as the Coherent Point Drift (CPD), GMMTree registration and L2 distance registration algorithms. The former's rigid version has been used for this study, due to its simplicity on the low-level. As such, an initial manual transformation and scaling was required to bring the two data-sets closer before the CPD algorithm took action. To manually scale the two datasets, information regarding the CT's voxel resoltion and the build's layer height were used.

The CPD algorithm outputs a rigid transformation matrix in the form of eq.(1)

$$\begin{bmatrix} X_s \\ Y_s \\ Z_s \\ 1 \end{bmatrix} = \begin{bmatrix} r11 & r12 & r13 & t_x \\ r21 & r22 & r23 & t_y \\ r31 & r32 & r33 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_t \\ Y_t \\ Z_t \\ 1 \end{bmatrix} \tag{1}$$

An example of registered pores detected from the CT scan and in-situ data collected from the monitoring system is visualised in Figure A1.



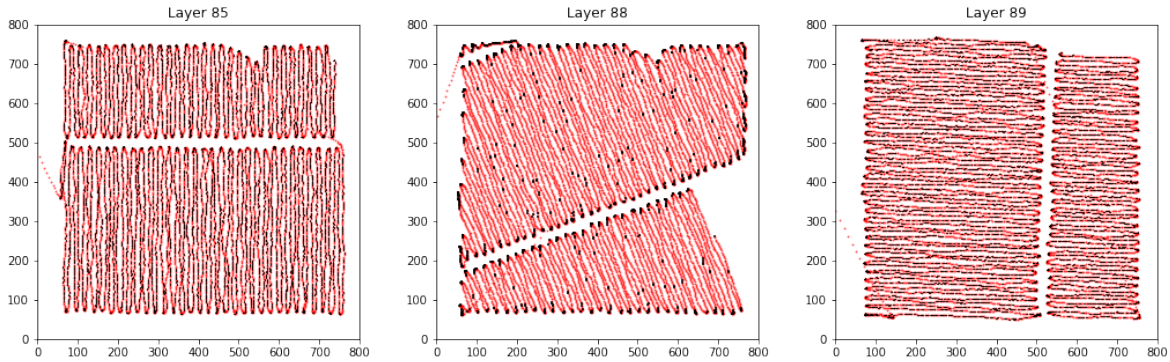
A1. Pore location w.r.t. optical data stream locations (layers 103-110) after registration. Pores with diameter $\leq 5\,\mu m$ are not visualised here.

2. Laser track turning point detection

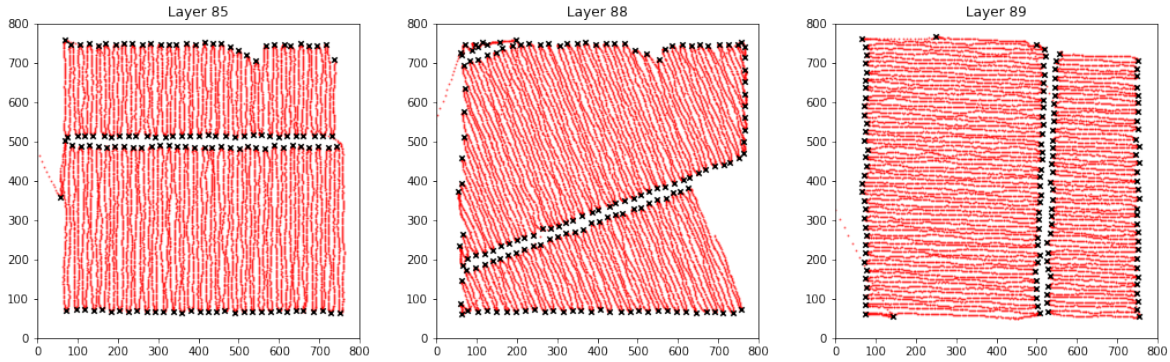
From the CT data it was evident that a large portion of the part's porosity occurred near its physical boundaries. This is commonly observed in AM parts, whose surface roughness can be relatively high when compared to other manufacturing methods. This has been explained in the background section. In addition, the available optical data stream did not include the frames associated with the finishing scans, which usually target the exterior of the part. As such, the optical coverage relating to the pores near the edges was considered incomplete. In order to tackle this issue, the pores and frames near the physical boundaries of the printed object were excluded from the analysis.

Given an average of 140 turning points for each of the 50 layers used for this study, more than 6e+3 turning points had to be identified and located. This would be a repetitive task if done manually - the task becomes almost intractable if one considers all 840 available layers. As such, an automatic turning point detection algorithm has been designed and implemented.

The main idea behind the algorithm is the calculation of the signed differences between the (x,y) coordinates of pairs of consecutive frames. The distances should be signed because the direction of change is crucial to the problem. The differences were stored in separate arrays for X and Y to avoid entanglement, proven to be useful in the following steps. The arrays were then passed to the sign operation; the indices of frames at which cross-over (sign change) occurs are detected and saved in separate lists. By plotting the detected cross-overs in both X and Y



(a) Unfiltered turning points



(b) Filtered turning points

A2. Turning point detector results for different hatching orientations. Turning points marked with 'x' and frames of layer marked with red dots.

arrays, two observations were made; (i) each actual turning point was associated with multiple frame indices and (ii) light noise was observed within the volume. By trial and error it was found that (i) can be mitigated if only the shortest array between X and Y is considered for the analysis and has been related to the hatch direction of the layer. Interestingly, for (ii), the observed noise was not an algorithmic error; instead, this could be an instrument error arising from the gimbals's operation, and hence, the regions of the layer associated with noise should be considered as points of interest. To isolate the turning points from the noise, the selected array was sorted and if the difference between consecutive indices was larger than a threshold (=5 via trial and error) the lower index was marked as noise.

The suggested algorithm results in 99.8% turn point detection accuracy, tested for all layers, and has been proven to be hatch-orientation invariant. It is also capable of detecting regions of interest in the print volume, where the laser scan speed is 0 m/s but not on a turning point.

Algorithm 1: Spherical proximity filter

clearance = x

for *pore in all_pores* **do**
  $threshold = pore.dim + clearance$
  for *frame in all_frames* **do**
    $distance = \sqrt{(x_{pore} - x_{frame})^2 + (y_{pore} - y_{frame})^2 + (z_{pore} - z_{frame})^2}$
    if $distance < threshold$ **then**
      $frame.label = 1$
    else
      $frame.label = 0$
    end
  end
end

Sample results can be seen in Figure A2, in which the importance of the suggested filtering step is also visualised.

3. Spatial proximity filter

Given accurate alignment between the two data-sets, the frames associated with each of the detected pores had to be identified. This helped assign binary labels to each available frame, '1' if the frame is within a given distance from any pore or '0' otherwise. Frames close to pores, especially keyhole pores which are typically larger and their formation is more apparent, are expected to have distinctive features like relatively large amounts of spattered material or melt pool spots with geometries whose shape falls outside the normal distributions.

To implement the proximity filter, a Euclidean distance function was implemented, measuring the distance between a pore's centroid and the coordinates of each available frame, before comparing it to a threshold equal to a specified clearance plus the spatial dimension of the pore itself. This can be visualised as a spherical boundary around the pore with diameter equal to $pore.dim + clearance$. Any frame falling within the boundary was labelled as 'bad' (represented by 1) while the remaining frames were labelled as 'good' (represented by 0). The function's pseudo-code can be found in Algorithm **??**.

It was noted that during the formation of a keyhole pore the laser penetration exceeds normal

levels due to high recoil pressure. Specifically, the penetration depth can reach up to 200 µm, and thus, a keyhole pore may be associated with frames located up to 5 layers above the reach of the spherical proximity filter. One way to address this would be to increase the clearance value, however, this would also lead to the inclusion of frames spanning along X and Y axes which may be completely irrelevant to the pore of interest. Instead, another geometrical filter was implemented which which checks which frames have coordinates lying inside an ellipsoid (or prolate spheroid) stretching around the pore's centroid. This enabled the expansion of the original sphere solely in the Z axis direction by adjusting the ratio between the major and minor axis of the ellipsoid. To compensate for the increased complexity of the algorithm, specific mathematical manipulations and geometric principles were used, based on this.