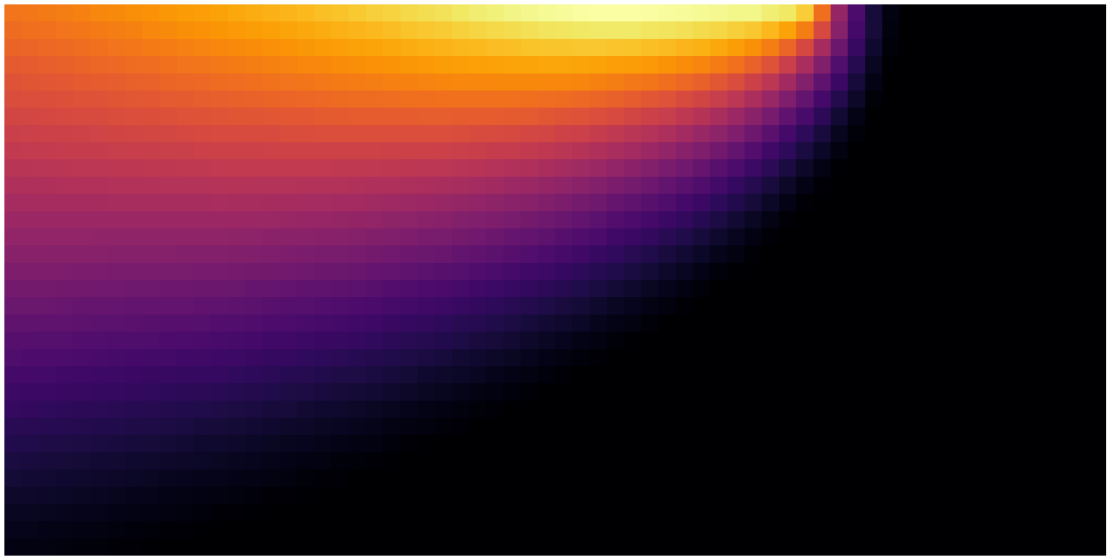

Physics-Informed Machine Learning for Thermal Field Prediction in Directed Energy Deposition

Master's thesis in Engineering Mathematics and Computational Science

Alexander Hughes

Department of Mathematics

CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2026
www.chalmers.se

Master's thesis 2026

Physics-Informed Machine Learning for Thermal Field Prediction in Directed Energy Deposition

Alexander Hughes

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Mathematics
Chalmers University of Technology
Gothenburg, Sweden 2026

Physics-Informed Machine Learning for Thermal Field Prediction in Directed Energy Deposition
Alexander Hughes

Supervisors: Anna Moretti (GKN Aerospace), Christopher Wulff (GKN Aerospace), Johan Jonasson (Chalmers)
Examiner: Johan Jonasson (Department of Mathematics), Chalmers University of Technology

Master's Thesis 2026
Department of Mathematics
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: Predicted melt pool temperature using a machine learning based surrogate model as a 2D slice.

Typeset in LaTeX
Printed by Chalmers Reproservice
Gothenburg, Sweden 2026

Physics-Informed Machine Learning for Thermal Field Prediction in Directed Energy Deposition
Alexander Hughes
Department of Mathematics
Chalmers University of Technology

Abstract

Directed Energy Deposition (DED) is an additive manufacturing process in which accurate thermal predictions are essential for understanding melt-pool behaviour. While high-fidelity simulations can provide this information, they are computationally expensive, especially when a large number of simulations are required. This motivates the development of faster predictive methods.
This thesis investigates the use of machine learning (ML) surrogate models as an efficient alternative for thermal prediction in DED processes. Both data-driven and physics-informed approaches are considered, including neural networks and neural operators. The models are trained on data generated from high-fidelity simulations, with the goal of evaluating trade-offs between predictive accuracy, computational cost and the incorporation of physical constraints.
A paired benchmark spans three architecture families, each with a data-only and a physics-informed variant, giving six model variants in total. Trained across four training-data fractions and three initialisation seeds, the data-only pointwise MLP achieves the best trade-off. It produces a root mean square error (RMSE) of 12.1 ± 0.3 K (approximately 0.7% of the liquidus temperature of 1928 K) on the in-distribution test set and 46.9 ± 16.6 K on the process-extreme out-of-distribution (OOD) split. At 8.6 ms per output - roughly $7\,000\times$ faster than the GPU-accelerated solver - the surrogate is sufficiently accurate for unlocking new use cases such as interactive process-parameter exploration, where rapid iteration matters more than solver-level precision. Physics-informed training through soft partial differential equation (PDE) and boundary losses does not improve overall accuracy and substantially increases training time, although it does provide some regularisation benefits at high temperatures on OOD cases. The volume-averaged RMSE is dominated by ambient voxels (87% of the domain below 400 K). Out-of-distribution generalisation remains the dominant limitation, with OOD error roughly four times larger than random-test error. These results suggest that for this fixed-domain setting, compact data-driven surrogates are preferable to more complex physics-informed or operator-based models. Future work should test whether these results hold at finer grid resolutions and explore uncertainty estimates to flag unreliable predictions.

Keywords: Directed Energy Deposition, Machine Learning, Physics-Informed Neural Networks, Thermal Prediction, Additive Manufacturing, Surrogate Models, Neural Operators, Out-of-Distribution Generalisation

Acknowledgements

I would like to thank my supervisors for their guidance and support throughout this project. At GKN Aerospace, Anna Moretti and Christopher Wulff provided invaluable insight into the industrial context of this work and made time for regular meetings despite demanding schedules. At Chalmers, Johan Jonasson offered steady academic supervision and constructive feedback at every stage.
I am grateful to my family and friends for their encouragement and patience during the long evenings and weekends that this thesis required. This work would not have been possible without their unwavering support.

Alexander Hughes, Gothenburg, June 2026

Contents

List of Figures

List of Tables

List of Acronyms

AM Additive Manufacturing. 1–6, 8

CNN Convolutional Neural Network. 12

DED Directed Energy Deposition. 1, 3–7, 13–15, 46
DL Deep Learning. 2

FEM Finite Element Method. 2, 9
FNO Fourier Neural Operator. 10, 11, 13–15, 21
FVM Finite Volume Method. 2, 3, 5, 7, 19, 37, 47

GP Gaussian Process. 12

IoU Intersection over Union. 26, 28, 30, 32

LPBF Laser Powder-Bed Fusion. 1, 4, 10, 12
LSTM Long Short-Term Memory. 12

ML Machine Learning. 2, 3, 9, 17
MLP Multi-Layer Perceptron. 20

PDE Partial Differential Equation. 8–10, 12, 19
PIML Physics-Informed Machine Learning. 2–6
PINN Physics-Informed Neural Network. 8–14, 20
PINO Physics-Informed Neural Operator. 3, 11–15

RMSE Root Mean Square Error. 12, 46

SSIM Structural Similarity Index Measure. 13, 16

Nomenclature

Below are the symbols used throughout this thesis, grouped by category. Dimensionless quantities and abstract operators are marked "-" in the unit column.

Symbol	Description	Unit
Fields and Coordinates		
$T(\mathbf{x}, t)$	Temperature field	K
$\hat{T}_\theta$	Surrogate-predicted temperature field	K
T_{FVM}	High-fidelity FVM reference temperature field	K
T_{amb}	Ambient temperature (293.15 K)	K
$\mathbf{x} = (x, y, z)$	Spatial coordinates (laboratory frame)	m
$\mathbf{r} = (\xi, y, z)$	Source-attached coordinates ($\xi = x - vt$)	m
t	Time	s
$\Omega,\ \partial\Omega$	Spatial domain and its boundary	m^3, m^2
$\partial\Omega_D$	Dirichlet portion of the boundary	m^2
Process Parameters		
$\mathbf{u}$	Process-parameter vector $(P, v, \dot{m}, d_{\text{spot}}, \dots)$	-
$\mathcal{U}$	Process-parameter space ($\mathbf{u} \in \mathcal{U}$)	-
P	Laser power	W
v	Scan speed	m/s
$\dot{m}$	Material feed rate	kg/s
d_{spot}	Laser spot diameter	m
α	Absorptivity	-
$\mathbf{p}_{\text{extra}}$	Additional process parameters (catch-all)	-
Material Properties		
ρ	Density	kg/m^3
c_p	Specific heat capacity	J/(kg·K)
k	Thermal conductivity	W/(m·K)
$\mathbf{m}$	Vector of material properties $(\rho, c_p, k, \dots)$	-
Governing Physics		
$Q(\mathbf{x}, t)$	Volumetric heat source term	W/m^3

Symbol	Description	Unit
Q_{laser}	Laser heat-input model	W
$\mathcal{N}[\cdot]$	Governing PDE operator (heat equation)	-
$\mathcal{B}[\cdot]$	Boundary-condition operator	-
$\mathcal{I}[\cdot]$	Initial-condition operator	-
f	PDE source term in the residual ($f = Q$)	$\mathrm{W/m^3}$
g	Prescribed boundary value	K
Surrogate and Machine-Learning Operators		
$\mathcal{G}$	True physical solution operator	-
$\mathcal{G}_\theta$	Learned surrogate operator	-
$\mathcal{O}_\theta$	Neural-operator architecture (FNO, DeepONet, PINO)	-
f_θ	Neural-network mapping / backbone	-
θ	Learnable model parameters	-
a	Operator input function ($a \in \mathcal{A}$)	-
$\mathcal{A},\ \mathcal{V}$	Input and output (solution) function spaces	-
σ	Nonlinear activation function	-
v_k	Hidden feature field at FNO layer k	-
W_k	Pointwise weight matrix (FNO layer)	-
R_k	Learned spectral filter (FNO layer)	-
$\mathcal{F},\ \mathcal{F}^{-1}$	Fourier transform and its inverse	-
$b_i,\ t_i$	DeepONet branch and trunk outputs	-
p	Number of branch/trunk basis functions	-
Loss Functions and Weights		
$\mathcal{L}$	Total loss function	-
$\mathcal{L}_{\text{data}}$	Data-misfit loss term	-
$\mathcal{L}_{\text{PDE}}$	Physics / PDE-residual loss term	-
$\mathcal{L}_{\text{BC}}$	Boundary-condition loss term	-
$\lambda_{\text{data}}, \lambda_{\text{PDE}}, \lambda_{\text{BC}}$	Weights for the data, PDE and boundary loss terms	-
Datasets and Sampling		
$N_d,\ N_r,\ N_b$	Number of data, interior and boundary collocation points	-

1
Introduction

1.1 Background

Additive Manufacturing (AM) - the layer-by-layer deposition of material, usually metal powder or wire - has seen widespread adoption over the last decade. Thanks to its ability to produce complex geometries in an efficient manner, AM is now a go-to approach for rapid prototyping and the manufacture of components with complex geometries. Parts produced using AM benefit from greatly reduced material waste, making the process an attractive option for aerospace components where a low "buy-to-fly ratio" ratio is desirable.
Aerospace-grade AM increasingly targets safety-critical engine infrastructure - turbine blades, fan casings and structural brackets - where accuracy and microstructural integrity are paramount. Current production workflows rely on Directed Energy Deposition (DED) for repair and large-scale builds and Laser Powder-Bed Fusion (LPBF) for high-resolution, intricate components. In this study, we will focus on DED because the melt pools produced by DED are larger than those produced by LPBF due to higher energy input, leading to slower cooling rates and stronger sensitivity of the thermal field to process parameters.

1.1.1 The Process Parameter Problem

Each AM setup - whether it uses wire or powder, DED or LPBF - is defined by a set of process parameters. These parameters control how the machine behaves and thus have a direct impact on the final outgoing part. They include spot diameter, heat input, material feed rate and scan speed. Changing one parameter often influences others. For example, increasing the feed rate might shorten the build time but it may also lead to defects unless other parameters are adjusted to mitigate these defects. This quickly becomes a complex problem. One might ask: what combination of process parameters yields the best result? Which combinations minimise defects while still meeting constraints such as build time? In this context, "best" refers to a combination that produces:

- a stable and robust process,
- a competitive deposition time,
- a minimal amount of defects, such as cracks, porosities and lack of fusion,
- suitable mechanical properties for the intended use-case.

Answering these questions usually requires extensive experimentation and analysis,

which is both costly and time-consuming. Because of these practical limitations, it becomes difficult to fully explore the design space, especially as parameter dimensionality and coupling increase.

1.1.2 Why Thermal Predictions Matter

The section above motivates the introduction and usage of design tools that can predict how different combinations of process parameters affect outcomes like the melt pool behaviour or defect formation. In particular, predicting the thermal field of the melt pool is essential, because it directly influences defect likelihood, microstructural properties and overall part quality.
Traditional simulations can provide accurate thermal predictions, but they are slow. This lack of speed compounds when many simulations are required. Simulations for the temperature field of the melt pool can take hours using Finite Volume Method (FVM) or Finite Element Method (FEM), which makes testing various combinations of process parameters wholly intractable. This motivates the introduction of faster surrogate models based on Machine Learning (ML) which can reduce the time required for predictions by orders of magnitude, provided they retain sufficient fidelity.
ML provides rapid inference at a fraction of the cost/time of traditional solvers. However, ML models are data-hungry and unfortunately data in AM is scarce. As mentioned earlier, high fidelity simulations may take hours each and experimental thermal measurements require expensive tools, severely limiting the collection of usable training data. As a consequence, the resulting training datasets for AM thermal prediction are relatively small when compared to other Deep Learning (DL) domains. These datasets consist of high fidelity simulations.
Physics-Informed Machine Learning (PIML) combats this by embedding physical laws - for example, a given heat equation - directly into the model. This is typically done by adjusting the loss function to incorporate these physical laws. This approach reduces the amount of labelled simulation, or experimental, data required to train PIML-based models. This feature makes PIML a natural choice for building a thermal surrogate for AM. In our case, the governing physics is the transient heat equation with a moving volumetric source, subject to boundary and initial conditions.
A particularly suitable direction is operator learning which has the goal of learning mappings between function spaces rather than focusing on a single solution. An operator-based approach allows prediction of the thermal field for new combinations of process parameters without retraining the model. In principle, this same framework extends to multiple materials as well, which would contribute to a scalable and flexible thermal surrogate capable of generalising across materials. In this thesis we evaluate this on a single material (Ti-6Al-4V), but the formulation is designed so that cross-material generalisation can be tested in future work. This addresses a key gap since many existing surrogates do not generalise well across materials, geometries or process windows under sparse data.

1.2 Problem Statement

The core question is whether we can build a surrogate model for the melt pool thermal field that is accurate and fast enough to support process-parameter design tasks.
Mathematically, we frame the problem as learning a parametric operator

$$\mathcal{G}_\theta : \underbrace{\mathbf{u} = \big(P,\, v,\, \dot{m},\, d_{\text{spot}},\, \dots\big)}_{\text{process-parameter vector}} \longmapsto \quad \underbrace{T(\mathbf{x},\, t)}_{\text{temperature field}} \quad , \tag{1.1}$$

where the *process-parameter vector* $\mathbf{u}$ collects the laser power P, scan speed v, material feed rate $\dot{m}$, laser spot diameter d_{spot} and any further descriptors $\mathbf{p}_{\text{extra}}$ (e.g. absorptivity, geometry); the material properties $\mathbf{m} = (\rho, c_p, k, \dots)$ are held fixed. This vector $\mathbf{u}$ is the unified parameter notation used throughout the thesis.
The operator $\mathcal{G}_\theta$ is parameterised by learnable weights θ and aims to satisfy:

1. **Accuracy.** Predictions should align with the high-fidelity FVM solutions within an acceptable tolerance.
2. **Speed.** The surrogate should be orders of magnitude faster than FVM to enable rapid exploration of the process-parameter space.
3. **Data efficiency.** The model must learn from a limited number of simulations (and even fewer experiments), ideally using physics to compensate for the lack of data.

This leads to several accompanying questions:

1. Which neural-operator architectures are most suitable for learning $\mathcal{G}_\theta$?
2. How much does embedding the governing heat equation into the loss improve accuracy, data efficiency and out-of-distribution behaviour?
3. What accuracy-cost trade-offs arise between physics-informed operators, instance-specific Physics-Informed Neural Networks (PINN)s and purely data-driven models?

In short, we target a surrogate that preserves the accuracy of high-fidelity models while enabling rapid design-space exploration and generalisation across materials and process windows.

1.3 Aim and Objectives

The goal of this thesis is to evaluate if PIML methods can be used as efficient and accurate surrogates for melt pool thermal prediction in DED AM. The work will have the following objectives:

- Produce a structured literature review of ML and PIML based methods for thermal prediction and identify any gaps in the current state of affairs.
- Implement a comparison of six model variants across three architecture families (spectral operators, branch-trunk networks, pointwise MLPs), each with data-only and physics-informed ablations. The physics-informed variants (Physics-

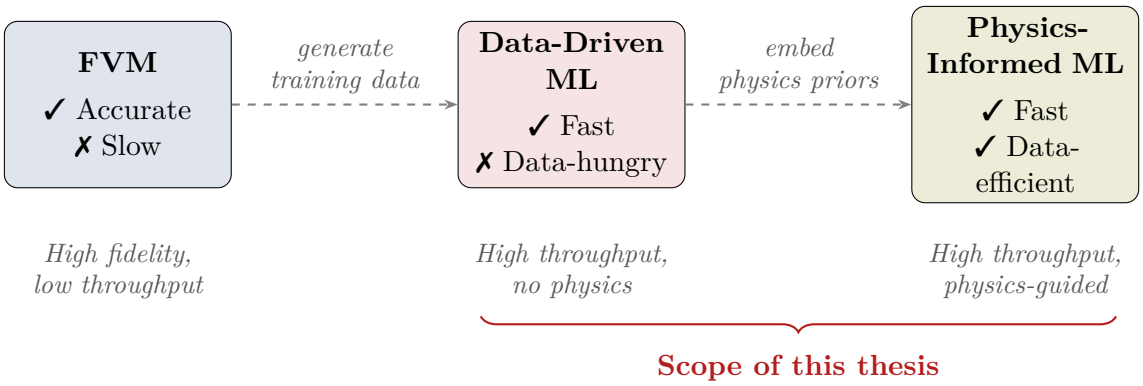

Figure 1.1: The motivation for this work stems from the fact that high-fidelity FVM provides accurate results but requires large datasets with no physical guarantees. Physics-informed operator learning combines fast inference with embedded physical constraints, requiring fewer training samples.

Informed Neural Operator (PINO), DeepONet+PDE, pPINN) are special cases of PIML.

- Generate a dataset of simulation cases of sufficient size to train and evaluate all six model variants.
- Implement the models in code using Python and evaluate them on accuracy, computational cost and data efficiency, using a consistent framework.
- Provide a recommendation and guidelines for those looking to deploy surrogates for AM thermal prediction based on our findings.

1.4 Scope and Limitations

In this thesis we focus on the application of PIML solely for predicting the thermal field of the melt pool. Other areas of interest such as defect detection, bead geometry and mechanical property prediction are out of scope for this thesis. In addition we will limit ourselves to:

- investigating DED only (LPBF based papers and resources may be used in the literature review),
- titanium and nickel-based superalloys used in production at GKN Aerospace, primarily Ti-6Al-4V,
- the available simulation data and in-house solver pipeline.

This scope isolates the thermal surrogate problem and avoids confounding factors from downstream phenomena (e.g., solidification microstructure, residual stress).

1.4.1 Use of Generative AI

Generative AI tools were used as writing assistants for drafting, rephrasing, brainstorming and generating plots/figures. All technical content, analysis, and conclusions are the author's own.

1.5 Contributions

In this thesis we will make the following contributions:

1. Controlled comparison of six model variants across three architecture families (spectral operators, branch-trunk networks, pointwise MLPs), each with data-only and physics-informed ablations.
2. A quantitative data-efficiency analysis of each method.
3. An analysis of the accuracy-cost-generalisability trade-offs of each model.
4. A temperature-stratified error analysis quantifying the ambient-bias in volume-averaged metrics and revealing where models actually differ in the thermally active region.
5. A quantitative demonstration that the best surrogate achieves approximately $7\,000\times$ speed-up over the GPU-accelerated FVM solver at inference, enabling interactive process-parameter exploration.
6. Practical recommendations and guidelines for selecting and implementing PIML surrogates based on the findings.

1.6 Overview

Chapter 2 provides a structured literature review mapping out current methods for thermal prediction for DED AM and identifying research gaps. Chapter 3 presents the methodology and model selection. Here we motivate the choice of each model, discuss implementation details and provide the evaluation framework. Chapter 4 presents the results and discussion from the final benchmark, organised thematically around the research questions. Chapter 5 provides practical recommendations and directions for future research.

2

Theory and Literature Review

2.1 Scope and Review Method

This chapter reviews thermal-prediction methods for metal AM, with emphasis on DED. The scope includes classical high-fidelity thermal modelling, data-driven surrogates, physics-informed networks and neural-operator approaches. The aim is to map out the methodological landscape and identify gaps that motivate the model selection and controlled comparison in Chapter 3. The review focuses on metal processes and excludes polymer AM unless a method is directly transferable to thermal prediction.
The literature was gathered iteratively: we began with seminal works on PIML and neural operators, then followed citation trails to DED-specific studies.

2.2 Thermal Modelling in DED

2.2.1 Why Thermal Modelling Matters

During DED, each newly deposited track is exposed to rapid heating and cooling as the laser, powder/wire stream, and substrate interact. This leads to a highly transient thermal history that strongly influences the resulting material behaviour. The thermal field controls the melt pool geometry, which sets bead shape, penetration, dilution and layer bonding - key factors for achieving dimensional accuracy and stable deposition. At the same time, the cooling rates and thermal gradients govern grain morphology, microstructure evolution, and the risk of defects such as lack-of-fusion or solidification cracking. Because these effects are thermally driven, even small errors in temperature prediction can propagate into larger inaccuracies when assessing microstructure, residual stresses or overall part quality. For DED, where heat accumulation and multi-layer interactions are pronounced, efficient thermal modelling is essential for robust process parameter design and optimisation.

2.2.2 The DED Process and Governing Physics

DED uses a focused energy source (laser or electron beam) to melt feedstock onto a substrate or previously deposited layer. In the powder-based variant considered here, metal powder is delivered through a coaxial nozzle and melted by the laser to form a melt pool that solidifies into a track as the head scans across the workpiece, as illustrated in Figure 2.1. This process is governed by the transient heat equation

with a moving top-hat source

$$\rho\, c_p \frac{\partial T}{\partial t} = \nabla \cdot \Big(k\, \nabla T \Big) + Q(\mathbf{x}, t), \tag{2.1}$$

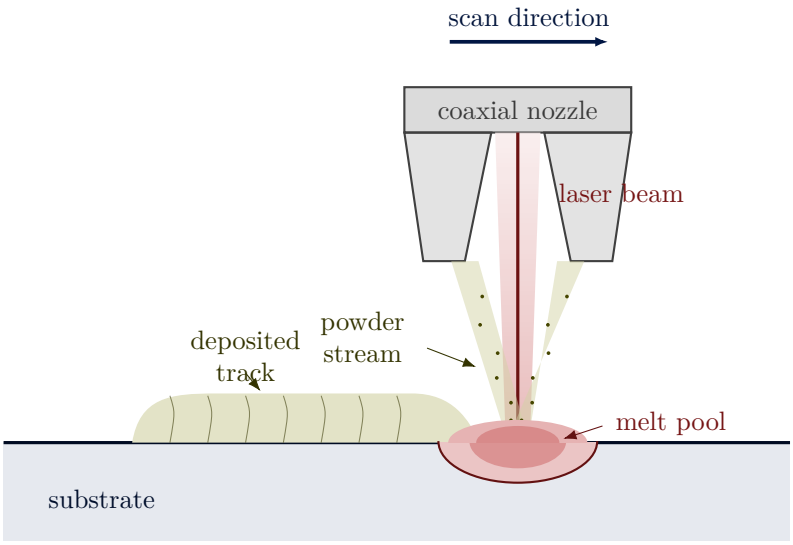

Figure 2.1: Schematic of the powder-based DED process. A laser delivered through a coaxial nozzle melts metal powder onto the substrate, forming a melt pool that solidifies into a deposited track as the head scans across the workpiece. The thermal field of the melt pool, the focus of this thesis, governs bead geometry, microstructure and defect formation.

where ρ is density, c_p specific heat capacity, k thermal conductivity and $Q(\mathbf{x}, t)$ the top-hat heat source. The most widely used source model is the Goldak double-ellipsoidal distribution [6]. Other beam models (e.g., Gaussian or uniform top-hat beams) can also be used depending on calibration data and required fidelity.
When temperature-dependent properties are used, the functions $k(T)$, $\rho(T)$ and $c_p(T)$ are evaluated at $T(\mathbf{x}, t)$ throughout the simulation.
Boundary conditions include convective and radiative losses as well as conductive heat transfer into the substrate. Key process parameters - laser power, scan speed, feed rate, spot diameter and absorptivity - enter through Q and through boundary-condition parameters such as emissivity and convection coefficients, which are treated as spatially uniform. This thesis focuses exclusively on powder-based DED.

2.2.3 Finite-Volume Modelling and Its Limits

FVM discretises Equation 2.1 to advance the temperature field in time on a control-volume mesh. Accuracy depends on mesh resolution near the heat source, stable implicit time stepping and calibrated boundary conditions.
Parametric studies - sweeping over possible process parameters - become intractable because each parameter setting requires an independent simulation. This motivates introducing fast surrogate models that provide thermal predictions in milliseconds once trained.

2.3 Physics-Informed Neural Networks

2.3.1 Foundations

Physics-Informed Neural Network (PINN)s, introduced by Raissi et al. [20], embed governing Partial Differential Equation (PDE)s directly into neural-network training through a physics-based loss term. This enables learning in data-scarce environments by enforcing consistency with physical principles.
A typical PINN minimises a composite loss:

$$\mathcal{L}(\theta) = \lambda_{\text{PDE}} \left\| \mathcal{N}[\hat{T}_\theta] - f \right\|^2 + \lambda_{\text{BC}} \left\| \mathcal{B}[\hat{T}_\theta] - g \right\|^2 + \lambda_{\text{data}} \left\| \hat{T}_\theta - T \right\|^2, \tag{2.2}$$

where $\mathcal{N}$ encodes the PDE, $\mathcal{B}$ the boundary/initial conditions, λ_{data}, λ_{PDE}, λ_{BC} denote the weights for the data loss, PDE loss and boundary loss respectively. For the transient heat equation, $f = Q(\mathbf{x}, t)$ is the volumetric heat source from the laser; g is the boundary condition; T is the reference temperature field from the high-fidelity simulation at observed data points; and $\hat{T}_\theta$ is the neural network's approximation of the temperature field (the surrogate), parameterised by trainable weights θ. Automatic differentiation supplies the spatial and temporal derivatives. In this work, Q is omitted from the residual (see Section 3.3), so the PDE loss enforces the source-free operator $\mathcal{N}[\hat{T}_\theta] \approx 0$.
PINNs are attractive for AM because thermal data are expensive to obtain. The PDE residual provides regularisation, improving physical consistency even with limited labelled data.
However, challenges include slow training on large 3D domains, sensitivity to loss-term weighting and difficulty resolving sharp gradients near the melt pool. Cuomo et al. [5] provide a detailed taxonomy of PINN variants.

2.3.2 Training Challenges

Loss balancing. The multi-term loss in Equation 2.2 creates a multi-objective optimisation problem. Imbalanced weights can cause one term to dominate. Remedies include augmented Lagrangian approaches [22], which dynamically adjust penalty multipliers to enforce PDE constraints, and adaptive loss-balancing methods [2, 9, 23], which automatically rescale loss-term weights during training to balance gradient magnitudes.

Residual mismatch. A further challenge arises when the PDE operator used in the physics loss (e.g., the quasi-steady-state heat equation with temperature-dependent properties) is a simplification of the actual data-generating process. If the reference fields (e.g., from a high-fidelity solver with temperature-dependent properties or calibrated boundary conditions) are not exact zero-residual samples under the simplified operator, forcing the neural network toward that residual can bias predictions away from the supervised target. This limits the effectiveness of physics-informed losses even when the adaptive weighting is well-tuned.

Spectral bias. Neural networks struggle to learn high-frequency components. In thermal fields this manifests as difficulty capturing steep gradients. Fourier-feature embeddings and multi-scale architectures help alleviate this issue.

Data scarcity. When data are extremely sparse, generative augmentation can help. For example, Generative Adversarial (GA)-PINN hybrids [11] synthesise physically plausible training samples.

2.3.3 Applications to AM Thermal Prediction

Xie et al. [24] predicted the 3D temperature field of a single-track Ti-6Al-4V deposit using a PINN, achieving 4.83% relative error compared to FEM. Training was significantly more efficient than a conventional ML model.
Zhu et al. [25] extended PINNs to predict temperature and melt pool fluid dynamics in metal additive manufacturing.
Peng et al. [19] predicted multi-layer thermal histories. Han et al. [7] integrated transfer learning to accelerate training for new parameter sets. Jiang et al. [10] demonstrated strong accuracy using less than 1% labelled data.

2.3.4 The Instance-Specificity Limitation

A trained PINN typically solves one fixed PDE instance: fixed process parameters, fixed boundary conditions. Changing laser power, scan speed or material properties generally requires retraining or fine-tuning. This makes PINNs unsuitable as parametric surrogates. Neural operators, by contrast, learn mappings from input functions (parameters, material properties) to solution fields.

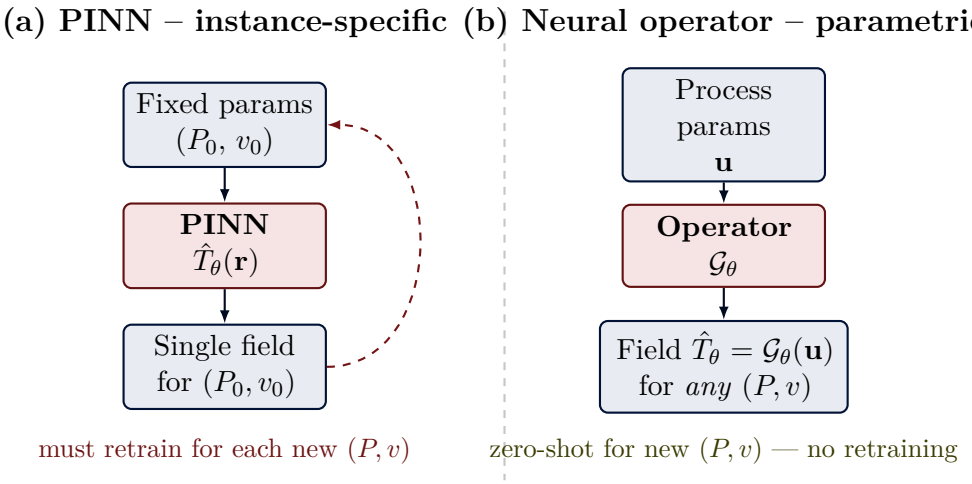

Figure 2.2: Conceptual comparison: **(a)** a PINN learns a single solution for fixed parameters and must be retrained for new conditions; **(b)** a neural operator learns the parametric map from the process parameters $\mathbf{u}$ to the temperature field, enabling predictions across the design space without retraining.

2.4 Neural Operators for Parametric PDE Surrogates

2.4.1 From Functions to Operators

Where PINNs learn a single function $\hat{T}_\theta(\mathbf{r})$ for one fixed PDE instance, neural operators learn mappings $\mathcal{G} : \mathcal{A} \mapsto \mathcal{V}$ between spaces of input and output functions. The input function $a \in \mathcal{A}$ encodes the problem-specific data that varies across instances - in general this could be spatially varying material properties, boundary conditions or a heat source distribution. *In this thesis the only quantities that vary across instances are the scalar process parameters*, so the input function reduces to the process-parameter vector $\mathbf{u} = (P, v, \dot{m}, d_{\text{spot}})$, broadcast as a spatially uniform field. We therefore treat a and $\mathbf{u}$ as the same operator input throughout; the concrete encoding for each architecture is given in Sections 2.4.3 and 2.4.2.

2.4.2 Deep Operator Networks (DeepONet)

DeepONet [14] consists of a branch network, which encodes the input function sampled at fixed sensor locations, and a trunk network that represents the spatial locations at which the output is evaluated. The network is given by

$$\mathcal{G}(a)(\mathbf{r}) = \sum_{i=1}^{p} b_i(a)\, t_i(\mathbf{r}). \tag{2.3}$$

where a denotes the input function, $b_i(a)$ are the branch network outputs encoding the discretised input function, and $t_i(\mathbf{r})$ are the trunk network outputs - feature coordinates evaluated at the query location $\mathbf{r}$. The final prediction is the inner product between branch and trunk outputs.
DeepONet has achieved strong performance across many PDE benchmarks but requires fixed sensor sampling, which limits flexibility with varying mesh resolutions or domains.
Safari and Wessels [21] applied a physics-informed DeepONet to thermal prediction in LPBF, and proposed sequential composition methods for multi-track prediction.

2.4.3 Fourier Neural Operator (FNO)

Fourier Neural Operator (FNO) [12] performs convolution in Fourier space

$$v_{k+1}(\mathbf{r}) = \sigma\Big(W_k v_k(\mathbf{r}) + \mathcal{F}^{-1}(R_k\, \mathcal{F}(v_k))\,(\mathbf{r})\Big), \tag{2.4}$$

where v_k is the hidden feature field at layer k, $\mathbf{r}$ is a specified spatial location, $\mathcal{F}$ is the Fourier transform, R_k is a learned complex-valued filter acting on the lowest frequency modes, and W_k is a local pointwise weight matrix. This offers discretisation invariance and natural super-resolution meaning the model can evaluate the solution on spatial grids different from those used during training. FNO excels with smooth solution fields but may underperform in regions with sharp gradients. FNO based models [1] have recently been shown to effectively map process parameters to 3D

temperature fields in laser welding, proving the validity of FNO as a surrogate model for this problem.

2.4.4 Physics-Informed Neural Operators (PINO)

PINO [13] augments neural-operator training with physics-based loss terms. Following the structure of the PINN loss (Equation 2.2), the composite loss is

$$\mathcal{L}_{\text{PINO}} = \mathcal{L}_{\text{data}} + \lambda_{\text{PDE}}\,\mathcal{L}_{\text{PDE}} + \lambda_{\text{BC}}\,\mathcal{L}_{\text{BC}}, \tag{2.5}$$

where $\mathcal{L}_{\text{data}}$ measures the mismatch between the operator's output and reference simulation data, $\mathcal{L}_{\text{PDE}}$ is the residual of the governing heat equation evaluated on the operator's output via automatic differentiation, and $\mathcal{L}_{\text{BC}}$ enforces boundary conditions. The scalar weights λ_{PDE} and λ_{BC} balance the physics terms against the supervised data term.

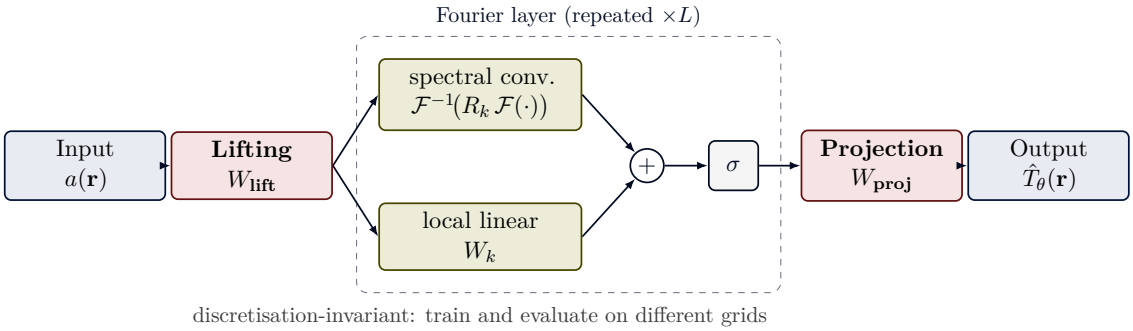

Figure 2.3: Fourier Neural Operator. The input function is lifted to a latent space, transformed to the spectral domain via the Fourier transform, convolved with learned spectral filters R_k, transformed back to physical space, and projected to the output. This gives discretisation invariance: the model can be trained on one grid resolution and evaluated on another.

2.4.5 Comparing Operator Architectures

DeepONet and FNO/PINO differ fundamentally. DeepONet works in the spatial domain with branch-trunk structure, while FNO works spectrally and supports mesh transfer and super-resolution. Concretely, this means that DeepONet learns the solution at fixed spatial locations, whereas FNO lifts the input function into Fourier space, where the learning instead occurs, before mapping back down to physical space; this makes inference independent of the discretisation used for training. PINO improves data efficiency by enforcing physics during training.
Chapter 3 implements both approaches alongside a baseline PINN and data-driven FNO for controlled comparison.

2.5 Data-driven and Hybrid Approaches

2.5.1 Convolutional Neural Networks

Convolutional Neural Network (CNN)s are a family of deep learning networks that learn features from inputs through convolutional operations (convolutional layers). These features are downsampled to reduce dimensionality (pooling layers) and then passed through fully connected layers to produce the final output. They can predict 2D/3D temperature fields on structured grids. Hemmasian et al. [8] built an LPBF thermal surrogate with $< 5\%$ relative Root Mean Square Error (RMSE). However, performance degrades substantially when material properties or boundary conditions vary.

2.5.2 Temporal and Recurrent Models

Thermal fields evolve over time due to layer-to-layer heat accumulation. CNN-Long Short-Term Memory (LSTM) hybrids [18] capture spatial-temporal dependencies. Implicit neural representations (INRs) - which model fields as continuous functions by mapping coordinates to network outputs - with Fourier features [15] offer continuous temperature fields with easy gradient extraction.

2.5.3 Gaussian Processes and Bayesian Methods

Gaussian Process (GP) regression provides calibrated uncertainty - outputs predictions in the form of probabilistic confidence intervals rather than point estimates - for low-dimensional thermal models. Mondal et al. [17] used GP surrogates with Bayesian optimisation to control melt pool dimensions. But scaling to high-dimensional spatio-temporal fields remains a challenge.

2.5.4 Multi-fidelity and Data Fusion

AM thermal modelling naturally offers multiple fidelity levels: analytical models (low) and thermal simulations (mid). Multi-fidelity learning [16] fuses these sources by training a low-fidelity model first, then learning a correlation mapping to high-fidelity targets. This can also be combined with neural operators (e.g., pretraining a PINO on coarse simulations and fine-tuning on high-fidelity data).

2.6 Synthesis

2.6.1 Limitations Observed Across Literature

1. **Instance specificity.** Most PINN-based surrogates focus on a single PDE instance and do not generalise across parameter sets.
2. **Limited controlled comparison.** Few studies compare neural operators, PINNs, and data-driven surrogates under identical conditions.

3. **Data-efficiency claims often unquantified.** Rigorous data-efficiency curves are rarely provided.
4. **Evaluation metrics.** Many works rely solely on scalar metrics. Field-level metrics (e.g., Structural Similarity Index Measure (SSIM), peak temperature error, melt pool boundary error) are underreported despite being essential for process design.
5. **Physics-data mismatch risk.** When the PDE operator embedded in the physics loss is a simplification of the actual data-generating process, the reference fields may not be exact zero-residual samples under that operator. If the solver produces data using temperature-dependent properties, calibrated boundary conditions, or empirical heat-source models that differ from the analytical residual, the physics term can become a source of bias rather than regularisation. This risk is discussed further in Section 2.3.2 and evaluated empirically in this study.

2.6.2 Design and Evaluation Framework

The literature review motivates several design principles:

1. **Parametric generalisation:** Prefer operator learning over PINNs for multi-parameter prediction.
2. **Physics-informed losses (hypothesis):** Embedding the governing PDE in the training loss is proposed to improve data efficiency and enforce physical consistency, particularly in data-limited regimes. This hypothesis is tested empirically in Chapter 4 by comparing physics-informed and data-only models across multiple training-set sizes.
3. **Controlled comparison:** Evaluate models on a shared dataset with identical metrics.
4. **Expanded metrics:** Include field-level metrics, data-efficiency curves, training cost and inference speed (how fast the trained model can output a prediction).

2.6.3 Research Questions

This study therefore asks:

1. How do physics-informed operator architectures (PINO, DeepONet+PDE) compare with data-driven operators (FNO, data-only DeepONet) and pointwise models (pPINN, Point MLP) for DED thermal prediction?
2. How does embedding the heat equation in the loss affect data efficiency, accuracy and out-of-distribution generalisation?
3. What are the cost-accuracy trade-offs of each method, and which is best suited for rapid process-parameter exploration?

Table 2.1 outlines the six model variants compared:

Table 2.1: Candidate models and their roles in this study.

Model	Family	Physics	Role
PINO	Neural operator (FNO backbone)	Yes (PDE loss)	Physics-informed spectral operator
FNO	Neural operator (FNO backbone)	No	Data-driven spectral operator
DeepONet+PDE	Neural operator (branch-trunk)	Yes (PDE loss)	Physics-informed branch-trunk operator
DeepONet	Neural operator (branch-trunk)	No	Data-driven branch-trunk operator
pPINN	Parametric neural network	Yes (PDE loss)	Physics-informed pointwise baseline
Point MLP	Pointwise neural network	No	Data-driven pointwise baseline

2.7 Summary

This chapter reviewed thermal modelling approaches in DED from classical solvers to data-driven surrogates, PINNs and neural operators. PINNs offer data efficiency but lack parametric flexibility. Neural operators (PINO, DeepONet) overcome this by learning mappings between function spaces and supporting generalisation across parameters. Data-driven approaches offer strong performance but lack physical guarantees. Pointwise models (pPINN, Point MLP) provide a middle ground, offering parametric flexibility at lower computational cost but requiring point-wise evaluation rather than full-field inference.

A clear gap exists: no prior work has systematically compared these six approaches - PINO, FNO, DeepONet+PDE, data-driven DeepONet, pPINN and Point MLP - on a shared DED dataset with both data-only and physics-informed ablations. Chapter 3 addresses this through a controlled experimental study.

3
Methodology

3.1 Overall Approach

This chapter outlines the methodology used in this study. The goal is to perform a controlled comparison of six model variants across three architecture families for melt pool thermal prediction in DED. The workflow consists of five main stages:

1. Collect the available simulation data and process constraints from GKN Aerospace.
2. Analyse the data structure, resolution, variability and practical constraints to define the modelling scope.
3. Incorporate insights from the targeted literature review (Chapter 2) to guide model and metric selection.
4. Implement six candidate models: data-only and physics-informed variants of spectral operators (FNO/PINO), branch-trunk networks (DeepONet), and pointwise MLPs (Point MLP/pPINN).
5. Evaluate all models using a unified framework with identical training data, metrics and testing protocol.

The experiment is structured as a paired ablation design across three architecture families. Within each family we compare a data-only baseline against a physics-regularised variant, so that the marginal effect of the PDE and boundary losses can be measured without confounding it with architecture changes. The full methodology pipeline is summarised in Figure 3.1.

3.2 Data and Constraints

In this section we investigate and detail the available data and constraints which may limit the study. The primary source of data in this study are thermal fields exported from an in-house solver built by GKN Aerospace. The final dataset consists of quasi-steady-state 3D finite volume solutions in a source attached frame and is denoted by $T(\mathbf{r}; \mathbf{u})$ where $\mathbf{u}$ is a given process parameter combination, and $\mathbf{r} = (\xi, y, z)$ are the source-attached coordinates (detailed in Section 3.3).

3.2.1 Inputs, Outputs and Parameter Space

The inputs in this study are given by quasi-steady-state 3D thermal fields where each sample is associated with a process parameter combination $\mathbf{u}$. This combination

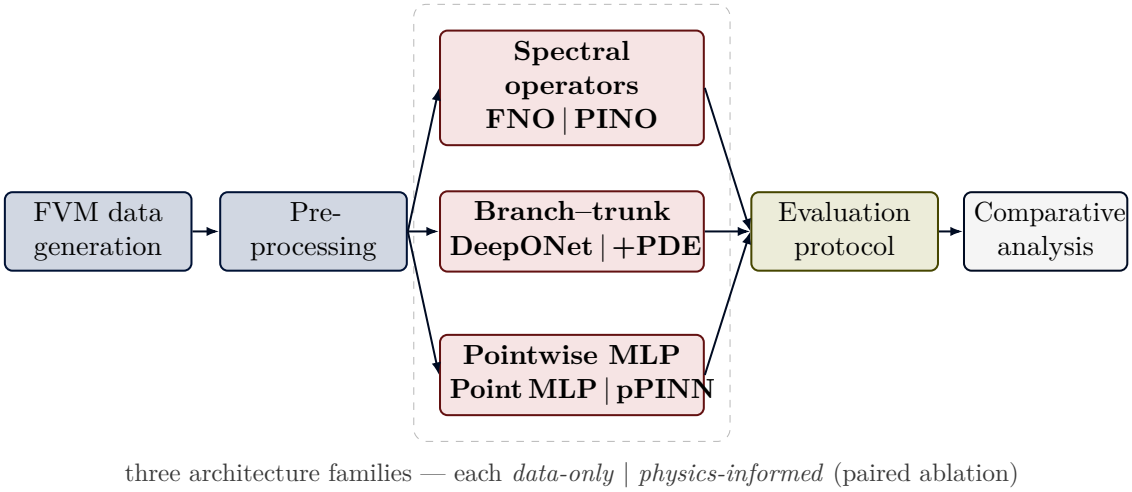

Figure 3.1: The end-to-end pipeline. FVM simulation data is generated, pre-processed into a common format, and used to train six model variants in three architecture-matched pairs. All models are evaluated with a shared metric suite and compared across accuracy, data efficiency, and computational cost.

includes power, scan speed, feed rate and spot diameter. Every sample will be interpolated onto a fixed 3D grid with equidistant spacing and dimensions $64 \times 32 \times 32$ (length ξ, height y, width z), covering a physical domain of 7 mm × 3 mm × 8 mm. The source-attached coordinates are defined in Section 3.3.
This study focuses on Ti-6Al-4V, an alloy used in production at GKN Aerospace. Temperature-dependent material properties are included as inputs to the models. All other parameters, or settings, are held constant in the simulations to ensure a balanced dataset. The process parameter combinations will be generated using a latin hypercube based approach with an additional filtering afterwards. Each generated combination will be required to fulfill certain constraints in order to be included in the experiment. This is to ensure that unrealistic process parameter combinations are removed and that we avoid polluting the training dataset with random noise. These constraints should also be respected when serving the model (i.e., deploying it for inference), to avoid training-serving skew caused by out-of-distribution queries.
The outputs in this study will be quasi-steady-state thermal fields over the full domain $T(\mathbf{r}; \mathbf{u})$ where $\mathbf{r} = (\xi, y, z)$ are the source-attached coordinates. Derivative properties of interest such as peak temperature and SSIM are computed directly from this field.

3.2.2 Preprocessing Inputs

Before we can train the candidate models we need a unified, common data representation suitable for all the aforementioned models. The raw FVM outputs are exported as VTU mesh files, which require transformation to a common learning format.
The output thermal field can be represented in two ways depending on the model architecture. Grid-based models (FNO and PINO) require the field to be on a fixed

3D voxel grid of size $64 \times 32 \times 32$ in the source-attached frame, interpolated from the FVM mesh. Pointwise models (Point MLP, pPINN) and branch-trunk networks (DeepONet) evaluate the field at sampled spatial coordinates over the same domain. In order to ensure consistency, the sampled points are taken directly from the grid representation so that all models operate over the identical physical domain.
The preprocessing pipeline can be defined by the following steps

1. choose the shared domain,
2. resolve the thermal field for each sample,
3. interpolate each sample onto a shared grid.

Note, that the shared grid remains constant throughout the experiment for training, validation and testing to ensure our input data is unified. When validating or testing the model, the shared bounds inferred from the training pipeline will be used. For the physics-informed models we also take care with the boundary of the shared domain to ensure it matches our PDE formulation. Concretely we ensure that the temperature at the relevant boundaries of the domain are constant at the chosen ambient temperature, which has the secondary effect of producing a slightly looser shared domain.

3.2.3 Practical Constraints

This study is shaped around a few practical constraints. The industrial data from GKN Aerospace is both finite and confidential, limiting the size of the training set and requiring obfuscation of the process parameter ranges and mesh statistics in this report. The second constraint is the available hardware, which determines the grid resolution and training duration of the ML models.

3.3 Problem Setup and Mathematical Formulation

To avoid repetition, the general problem formulation shared by all six candidate models is introduced here; Section 3.4 then specialises it to each architecture family. As mentioned in Chapter 2 the DED process is governed by the transient heat equation with a moving top-hat heat source given by

$$\rho\, c_p \frac{\partial T}{\partial t} = \nabla \cdot \Big(k\, \nabla T\Big) + Q(\mathbf{x}, t), \tag{3.1}$$

where ρ is density, c_p specific heat capacity, k thermal conductivity and $Q(\mathbf{x}, t)$ the top-hat heat source. The temperature-dependent properties - when used - transform the parameters ρ into $\rho(T)$ and they are evaluated at $T(\mathbf{x}, t)$. In our case we will be using temperature-dependent properties so alongside $\rho(T)$ we will have $k(T)$ and $c_p(T)$.
Since we are working with quasi-steady-state data we transform Equation 3.1 into a coordinate system moving with the heat source. Given this let $\mathbf{r} = (\xi, y, z)$ denote the source-attached coordinates where ξ is the longitudinal coordinate along the scan path (positive in the direction of laser travel), y is the transverse coordinate

perpendicular to the scan path, and z is the depth coordinate measured downward from the top surface.

$$\xi = x - vt$$

where v is the scan speed in the x-direction and t is time. In this moving frame the temperature field becomes independent of time ($\frac{\partial T}{\partial t} = 0$). In our shifted coordinate field we get, using the chain rule,

$$\frac{\partial T(\xi(x,t),t)}{\partial t} = \frac{\partial T}{\partial \xi}\frac{\partial \xi}{\partial t} + \frac{\partial T}{\partial t} = -v\frac{\partial T}{\partial \xi}$$

which when substituted into Equation 3.1 gives

$$\rho\, c_p v \frac{\partial T}{\partial \xi} + \nabla \cdot (k\, \nabla T) + Q = 0. \tag{3.2}$$

Provided k is constant this could be reduced to

$$\rho\, c_p v \frac{\partial T}{\partial \xi} + k\nabla^2 T + Q = 0, \tag{3.3}$$

however since we use temperature-dependent conductivity $k = k(T)$, we use Equation 3.2 as our governing PDE equation.
The computational domain is

$$\Omega = [\xi_{\min},\, \xi_{\max}] \times [y_{\min},\, y_{\max}] \times [z_{\min},\, z_{\max}], \tag{3.4}$$

which is fixed across all simulations and predictions. On the Dirichlet boundaries of the domain (sides, bottom, and trailing face) we impose fixed ambient temperature:

$$T = T_{\text{amb}} \quad \text{on } \partial\Omega_D, \tag{3.5}$$

where $T_{\text{amb}} = 293.15$ K. The top surface, where the laser heat source acts, is left without a Dirichlet constraint so that the temperature profile is determined by the heat input.
The boundary condition is denoted compactly as

$$\mathcal{B}[T] = T - T_{\text{amb}} = 0 \quad \text{on } \partial\Omega_D, \tag{3.6}$$

where $T_{\text{amb}} = 293.15$ K is the ambient temperature enforced on the Dirichlet portion of the boundary.
The goal of this study is to learn a surrogate $\hat{T}_\theta : \Omega \times \mathcal{U} \to \mathbb{R}$ that approximates the true temperature field T across the spatial domain Ω for any process-parameter vector $\mathbf{u} \in \mathcal{U}$. The learnable parameters θ are obtained by minimising a total loss function composed of three terms, summarised in Figure 3.2:

Data loss. This supervised term measures the mean squared error between the surrogate prediction and the FVM reference data on a set of N_d collocation points:

$$\mathcal{L}_{\text{data}} = \frac{1}{N_d}\sum_{i=1}^{N_d}\left\|\hat{T}_\theta(\mathbf{r}_i; \mathbf{u}_i) - T_i\right\|^2, \tag{3.7}$$

where $\mathbf{r}_i = (\xi_i, y_i, z_i)$ is a spatial location and T_i is the corresponding FVM temperature for process parameters $\mathbf{u}_i$.

PDE residual loss. For the physics-informed models this term enforces compliance with the quasi-steady-state heat equation (Equation 3.2) on N_r interior collocation points:

$$\mathcal{L}_{\text{PDE}} = \frac{1}{N_r} \sum_{i=1}^{N_r} \left\| \mathcal{N}[\hat{T}_\theta; \mathbf{u}_i](\mathbf{r}_i) \right\|^2, \tag{3.8}$$

where $\mathcal{N}$ is the source-free PDE operator evaluated on the model's prediction:

$$\mathcal{N}[\hat{T}_\theta] = \rho(\hat{T}_\theta)\, c_p(\hat{T}_\theta)\, v\, \frac{\partial \hat{T}_\theta}{\partial \xi} + \nabla \cdot \Big(k(\hat{T}_\theta)\, \nabla \hat{T}_\theta \Big), \tag{3.9}$$

i.e. the left-hand side of Equation 3.2 with the heat-source term Q omitted. The heat source term Q is omitted because the FVM data include dynamic laser-bead interaction that the fixed-source quasi-steady-state model cannot capture. For example, at high feed rates a large bead builds up on the substrate and intercepts the laser, altering the energy deposition profile. Since we only observe the substrate temperature field, modelling Q would require knowledge of the unknown surface geometry. The source-free residual is therefore only approximately correct away from the heat source where $Q \approx 0$. The residual is computed at each collocation point via automatic differentiation of $\hat{T}_\theta$ with respect to its spatial inputs.

Boundary-condition loss. This term penalises deviation from the boundary conditions (Equation 3.10) on N_b boundary points:

$$\mathcal{L}_{\text{BC}} = \frac{1}{N_b} \sum_{i=1}^{N_b} \left\| \mathcal{B}[\hat{T}_\theta; \mathbf{u}_i](\mathbf{r}_i) \right\|^2. \tag{3.10}$$

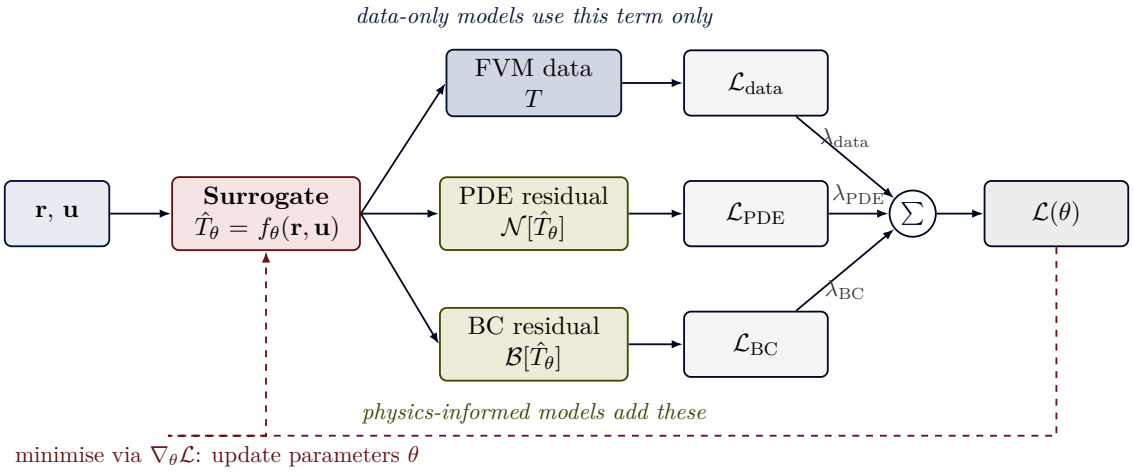

Figure 3.2: Composition of the training loss. The surrogate $\hat{T}_\theta$ is supervised by a data term against the FVM reference, while the physics-informed variants add a PDE-residual term $\mathcal{N}[\hat{T}_\theta]$ and a boundary term $\mathcal{B}[\hat{T}_\theta]$, combined with weights λ_{data}, λ_{PDE} and λ_{BC}. Data-only models use the data term alone.

3.4 Model Descriptions

The six candidate models form three architecture families, each pairing a data-only model with a physics-informed variant that adds the PDE-residual and boundary terms of Section 3.3. We introduce them family by family; full network and training settings are deferred to Section 3.5.

3.4.1 Pointwise MLP family: Point MLP and pPINN

The pointwise family treats prediction as a regression over the joint spatial-parameter domain rather than as operator learning. A fully-connected Multi-Layer Perceptron (MLP) f_θ takes a single query location concatenated with the process parameters $\mathbf{u}$ and returns the temperature at that point,

$$\hat{T}_\theta(\xi, y, z; \mathbf{u}) = f_\theta((\xi, y, z), \mathbf{u}) \approx T(\xi, y, z; \mathbf{u}) \tag{3.11}$$

where the input vector has dimension $3 + \dim(\mathbf{u})$. Because the parameters are passed as ordinary inputs, a single trained network generalises across the parameter space without retraining, making this a parametric-but non-operator-baseline for testing whether operator learning adds value beyond simple parameter conditioning. Unlike an operator, which outputs the whole field in one forward pass, a pointwise model must be queried at every spatial location independently to reconstruct the 3D field, which is its main inference-cost drawback.

Point MLP (data-only). The data-only member of the family is trained with the supervised data loss alone (Equation 3.7). It is the simplest parametric baseline and the data-only counterpart against which the physics-informed pPINN is measured.

pPINN (physics-informed). pPINN [4] keeps the same network but augments the objective with the PDE-residual and boundary terms,

$$\mathcal{L}(\theta) = \lambda_{\text{data}}\mathcal{L}_{\text{data}} + \lambda_{\text{PDE}}\mathcal{L}_{\text{PDE}} + \lambda_{\text{BC}}\mathcal{L}_{\text{BC}}, \tag{3.12}$$

with the weights adapted during training as described in Section 3.5.3. It is the parametric extension of the classical PINN of Raissi et al. [20]: a standard PINN solves a single PDE instance and must be retrained for new parameters, whereas the parametric form passes the process parameters as inputs and removes that requirement, giving a fair physics-informed counterpart to the operator models.

3.4.2 Spectral operators: FNO and PINO

The Fourier Neural Operator [12] is a specialised neural operator. It will act as the data-driven operator baseline in our study. FNO has two attractive properties which make it a strong candidate model for our study; it is highly efficient thanks to the usage of the Fourier space, and it allows for zero-shot super-resolution meaning it is not limited to producing outputs on the grid defined by the training data. More concretely, this means that we can train this model on data with a coarse

resolution and still obtain high-resolution predictions thanks to the continuous output guaranteed by the neural operator architecture.
The model predicts the entire 3D temperature field in a single forward pass. Since the quasi-steady-state assumption eliminates the time dimension, there is no time channel in the output. FNO is preferred over CNN as a data-driven baseline because the comparison against PINO isolates the effects of the physics regularisation instead of architectural differences.
Neural operators rely on an iterative update not dissimilar to traditional neural networks, however since neural operators map functions to functions they differ slightly. Each layer is comprised of a non-linear activation function that acts on a linear integral operator with the first layer being a lifting layer to a higher dimensional space, and the last layer being a projection of the function back to the original function space. The idea behind this approach is to recognise that integral transforms - mapping a function from its original space into another - let us manipulate functions more efficiently in the new space and then map the result back using the inverse transform. Integral transforms such as the Laplace transform make this concrete: they turn differentiation into multiplication, which is exactly what makes them so effective for linear differential equations. The FNO uses the closely related Fourier transform.

Definition 3.4.1 (Fourier Transform)**.** Let $f \in L^1(\mathbb{R})$. The Fourier transform of f is defined by

$$\hat{f}(\xi) = \int_{-\infty}^{\infty} f(x)\, e^{-2\pi i x \xi}\, dx,$$

where $\hat{f} : \mathbb{R} \to \mathbb{C}$.

The key innovation of the FNO lies in its replacement of the linear part of the kernel integral operator used in traditional neural operators with a Fourier-based formulation which is made possible by assuming the kernel integral operator is translation invariant. Due to this assumption the integral operator can be represented as a convolution and thus efficiently computed in Fourier space.
The FNO parametrises a nonlinear solution operator

$$\mathcal{G}_\theta : a(\mathbf{r}) \mapsto T(\mathbf{r}). \tag{3.13}$$

where $\mathbf{r} = (\xi, y, z)$ is the spatial coordinate in the source-attached frame. The input function $a(\mathbf{r}) \in \mathbb{R}^4$ is a four-channel field on the grid in which each channel corresponds to one process parameter (power, scan speed, feed rate, spot diameter) broadcast uniformly across all spatial locations. The operator maps this input function to the corresponding output temperature field.
The first layer in the FNO is a lifting layer which lifts the input function to a higher-dimensional latent space. Afterwards a sequence of Fourier layers are applied until the final layer which projects back down to the output space. Each Fourier layer performs convolution in Fourier space and combines this with a local linear transformation in physical space

$$v_{k+1}(\mathbf{r}) = \sigma\Big(W_k v_k(\mathbf{r}) + \mathcal{F}^{-1}(R_k\, \mathcal{F}(v_k))\,(\mathbf{r})\Big), \tag{3.14}$$

where $\mathcal{F}$ is the Fourier transform, R_k is a learned complex-valued filter acting on the lowest frequency modes, and W_k is a local pointwise weight matrix.
The FNO is trained with the data loss alone (Equation 3.7), serving as the data-only baseline for PINO.

PINO (physics-informed). PINO is closely related to FNO in so far as it shares the same backbone (FNO). The novelty of PINO is the embedding of physics into the training loss which allows for more efficient training when compared to FNO. In principal PINO can have unlimited training data because we can sample any new input from any number of virtual PDE instances. In theory it can be trained solely with these sampled virtual points, however the training data from simulations or experiments provides stronger guarantees compared to physics constraints which makes optimisation easier. The power of PINO lies in its ability to combine data with physics constraints resulting in faster and more accurate neural operators.
PINO retains the same operator parametrisation $\mathcal{G}_\theta$ defined in Equation 3.13 and the same Fourier layers given by Equation 3.14, but augments the training objective with a physics-based residual loss derived from the governing PDE.
The PINO loss is defined as

$$\mathcal{L}_{\text{PINO}}(\theta) = \mathcal{L}_{\text{data}} + \lambda_{\text{PDE}}\,\mathcal{L}_{\text{PDE}} + \lambda_{\text{BC}}\,\mathcal{L}_{\text{BC}}, \tag{3.15}$$

3.4.3 Branch-trunk operators: DeepONet and DeepONet+PDE

DeepONet [14] takes a fundamentally different approach to learning operators compared to FNO. Instead of transforming the input function into a spectral representation, DeepONet utilises a branch-trunk architecture grounded in the universal approximation theorem [3] to learn operators accurately from limited amounts of data.
The architecture contains two subnetworks, shown in Figure 3.3. The trunk network is an MLP that encodes the spatial coordinates at which the output is evaluated, and the branch network is an MLP that encodes the process parameters (spatially averaged across the domain). In our implementation both have 3 hidden layers, width 64 and SiLU activations. Our implementation follows this parametric variant rather than the classical formulation in which the branch encodes a discretised input function.
In the standard formulation DeepONet computes

$$\mathcal{G}(a)(\mathbf{r}) = \sum_{i=1}^{p} b_i(a)\, t_i(\mathbf{r}), \tag{3.16}$$

where the branch network $b_i(a)$ encodes a discretised input function and the trunk network $t_i(\mathbf{r})$ encodes query spatial coordinates, and the prediction is their dot product. Our implementation follows a parametric variant: the branch input is the process-parameter vector $\mathbf{u}$ rather than a spatial function. Concretely, the input tensor contains spatial coordinates (ξ, y, z) and per-grid-point process parameters; the trunk network processes the spatial coordinates at each grid point, the parameter channels are spatially averaged and fed into the branch network, and the branch and

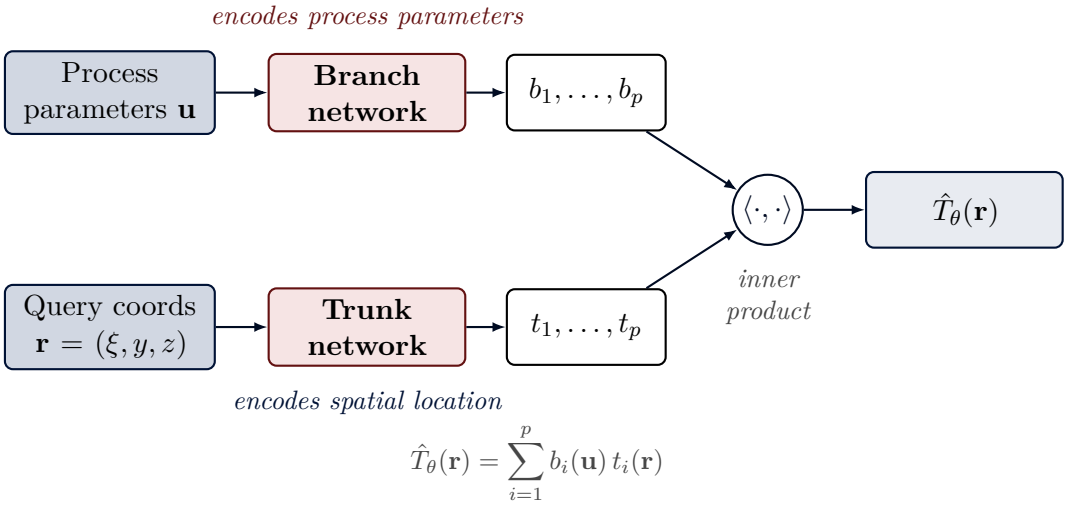

Figure 3.3: The DeepONet branch-trunk architecture. The branch network encodes the process parameters $\mathbf{u}$ and the trunk network encodes the query coordinates $\mathbf{r} = (\xi, y, z)$; their outputs are combined through an inner product to produce the predicted temperature $\hat{T}_\theta(\mathbf{r})$.

trunk outputs are combined via element-wise multiplication followed by a final linear projection layer.

DeepONet (data-only). The data-only DeepONet is trained with the supervised data loss $\mathcal{L}_{\text{data}}$ alone (Equation 3.7), ensuring a consistent comparison with the other operator models.

DeepONet+PDE (physics-informed). The physics-informed variant (inspired by the π-DeepONet framework of Lu et al.) augments the data loss with the same physics-based residual terms,

$$\mathcal{L}_{\text{DeepONet+PDE}}(\theta) = \mathcal{L}_{\text{data}} + \lambda_{\text{PDE}}\, \mathcal{L}_{\text{PDE}} + \lambda_{\text{BC}}\, \mathcal{L}_{\text{BC}}, \tag{3.17}$$

resulting in a physics-informed DeepONet variant.
The three physics-informed models-pPINN, PINO, and DeepONet+PDE-share a common training paradigm in which the total loss is a weighted sum of a supervised data term, a PDE residual term, and (where applicable) a boundary condition term. Figure 3.2 illustrates how the surrogate prediction feeds into each loss component and how the weighted combination forms the optimisation objective.

3.5 Implementation

All six model variants are trained and evaluated within a unified Python-based framework so that data preparation, preprocessing, training splits and evaluation metrics are shared. The goal of the framework is to help us isolate the differences in performance between the architectures by minimising any differences that could occur otherwise.

3.5.1 Computational Environment

The experiments were run on the following hardware

- GPU: NVIDIA Quadro P5200 16GB VRAM

and we utilise the following software stack

- Python
- PyTorch with CUDA backend

3.5.2 Model Implementation Details

All six models share a common output transform which is a softplus activation with floor value 0 and inverse temperature $\beta = 50$, to ensure we get non-negative temperature predictions. As mentioned earlier the spatial grid resolution is $64 \times 32 \times 32$ in the source-attached coordinate frame.

FNO and PINO. Both spectral-operator models use a 3D Fourier Neural Operator backbone with 3 Fourier layers, 20 hidden channels, 6 retained Fourier modes, and GELU activations. The FNO is trained with the data-only loss, while PINO adds the PDE and boundary residual terms described below.

DeepONet and DeepONet+PDE. The branch-trunk architecture uses a branch network and a trunk network, each with 3 hidden layers and hidden width 64, and SiLU activations. The element-wise product of branch and trunk outputs is projected through a final linear layer. The physics-informed variant (DeepONet+PDE) adds the same PDE and boundary losses as PINO.

Point MLP and pPINN. The pointwise MLP family uses a fully-connected network with 4 hidden layers, width 80 and SiLU activations. The input consists of the spatial coordinates (ξ, y, z) concatenated with the process parameters. The pPINN variant adds the PDE and boundary residual terms.

Physics residual computation. The PDE residual for the physics-informed models (PINO, DeepONet+PDE, pPINN) is computed using second-order central finite differences on the fixed grid, with temperature-dependent material properties interpolated from the solver's material property table. The heat source term Q is not included in the residual, so the PDE loss enforces the source-free operator $\mathcal{N}[\hat{T}_\theta] \approx 0$. The boundary condition loss enforces ambient temperature $T_{\text{amb}} = 293.15$ K on the Dirichlet boundaries. The implications of leaving out Q and the behaviour of the adaptive weight scheduler are discussed later in Chapter 4.

3.5.3 Training Protocol

All models share the same epoch-based training protocol. A single train/validation/test split is held fixed across architectures, and the optimisation schedule and early-stopping criteria are identical.

Optimiser and learning rate. Models are optimised with Adam using a learning rate of $\mathrm{lr} = 1 \times 10^{-3}$ and default moment decay rates $(\beta_1, \beta_2) = (0.9, 0.999)$. No weight decay is applied.

Batching. Because of GPU-memory constraints, operator-based models (FNO, PINO, DeepONet) are trained with a spatial batch size of 1 sample per iteration.

Epoch budget and early stopping. Training runs for at most 1,000 epochs. Validation loss is evaluated every 10 epochs; training is terminated early when the validation loss has not improved by at least $\min_\delta = 10^{-6}$ for 50 consecutive evaluation windows.

Loss weights. For the physics-informed models the total loss is

$$\mathcal{L}_{\text{total}}(\theta) = \lambda_{\text{data}}\,\mathcal{L}_{\text{data}} + \lambda_{\text{PDE}}\,\mathcal{L}_{\text{PDE}} + \lambda_{\text{BC}}\,\mathcal{L}_{\text{BC}}, \tag{3.18}$$

with initial weights $\lambda_{\text{data}} = 1.0$, $\lambda_{\text{PDE}} = 10^{-6}$ and $\lambda_{\text{BC}} = 10^{-2}$. To stabilise training, the physics weights undergo a linear warm-up over the first 100 epochs (scaling from 0 to their target values). After warm-up, an adaptive-weight scheme adjusts λ_{PDE} and λ_{BC} every 5 epochs so that the PDE and boundary losses track 2% and 5% of the data loss respectively, measured via an exponential moving average (decay 0.9). Updates are rate-limited to a factor of 1.25 per step and clipped to $\lambda_{\text{PDE}} \in [10^{-9}, 10^{-4}]$ and $\lambda_{\text{BC}} \in [10^{-2}, 10^{2}]$.

Reproducibility. Fixed random seeds are used for the train/validation/test split, weight initialisation, and stochastic sampling during training. The full study repeats each configuration across three seeds.

3.5.4 Evaluation Protocol

All models are evaluated on the same fixed data splits with a shared domain for the output predictions. Recall that under the quasi-steady-state assumption the temperature field is stationary in the source-attached frame, so evaluation is on spatial fields only - there is no time dimension. Let T denote the true temperature field and $\hat{T}$ the model prediction, both on the shared evaluation grid.

Primary accuracy metrics. The main metric is RMSE

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{k=1}^{n}(T^{(k)} - \hat{T}^{(k)})^2}, \tag{3.19}$$

where n is the number of grid voxels. It gives a single, interpretable measure of prediction error in Kelvin. MAE is also reported:

$$\mathrm{MAE} = \frac{1}{n}\sum_{k=1}^{n}|T^{(k)} - \hat{T}^{(k)}|, \tag{3.20}$$

which provides an error measure less sensitive to outliers than RMSE. Relative L^2 error $\varepsilon_{\text{rel}}^{L^2} = \|T - \hat{T}\|_{L^2(\Omega)} / \|T\|_{L^2(\Omega)}$ provides a scale-invariant measure of global fidelity. The per-case maximum absolute error $\max_{\mathbf{x}\in\Omega} |T(\mathbf{x}) - \hat{T}(\mathbf{x})|$ captures local failures that are invisible in aggregate RMSE.

Structural fidelity. SSIM is computed on three orthogonal slices (mid-depth, front face, and trailing face) to assess how well the predicted field preserves the spatial structure of the true field, rather than just matching pointwise values.

Melt-pool capture. Melt-pool Intersection over Union (IoU) is computed by thresholding both true and predicted fields at the Ti-6Al-4V liquidus temperature (1928 K) and measuring the intersection-over-union of the resulting binary masks. This gives a direct measure of how well the model reproduces melt-pool geometry, which is an important quality for process design.

Peak temperature. Hot-zone RMSE is evaluated on voxels where the true temperature exceeds the 95th percentile, probing accuracy in the high-temperature region near the melt pool.

Temperature-stratified RMSE. Because the majority of voxels in each case are at ambient temperature (approximately 87% below 400 K on the random-test split), the volume-averaged RMSE is dominated by the physically trivial region. To understand the errors in the thermally active region, voxels are also binned by their target temperature into nine intervals (293-400, 400-500, 500-600, 600-800, 800-1000, 1000-1200, 1200-1500, 1500-1928, and $\geq$1928 K) and per-bin RMSE is computed.

Splits. The 475 cases are split into training (50%), validation (20%), in-distribution test (15%) and out-of-distribution (15%) sets. Validation RMSE is used for early stopping while test and OOD RMSE are computed post-training. A single partition is fixed across all models and data fractions. The in-distribution split tests interpolation within the training parameter range. The OOD split tests extrapolation to process extremes - bridging the gap between simulation and experimental data, where process-edge cases are expensive and scarce. The OOD set comprises the 71 cases farthest from the training centroid in normalised parameter space (Euclidean distance across power, scan speed, feed rate and spot diameter, normalised using the full dataset statistics). Figure 3.4 shows the resulting split sizes together with the progressive training-data fractions used for the data-efficiency study.

Data efficiency. Each model is retrained on 10%, 25%, 50% and 100% of the available training data while keeping the validation and test sets fixed. This reveals whether physics-informed training is beneficial in low-data regimes and whether operator learning advantages emerge only at larger data fractions.

Computational cost. We record training wall-clock time, per-case inference latency, training time and the total number of trainable parameters.

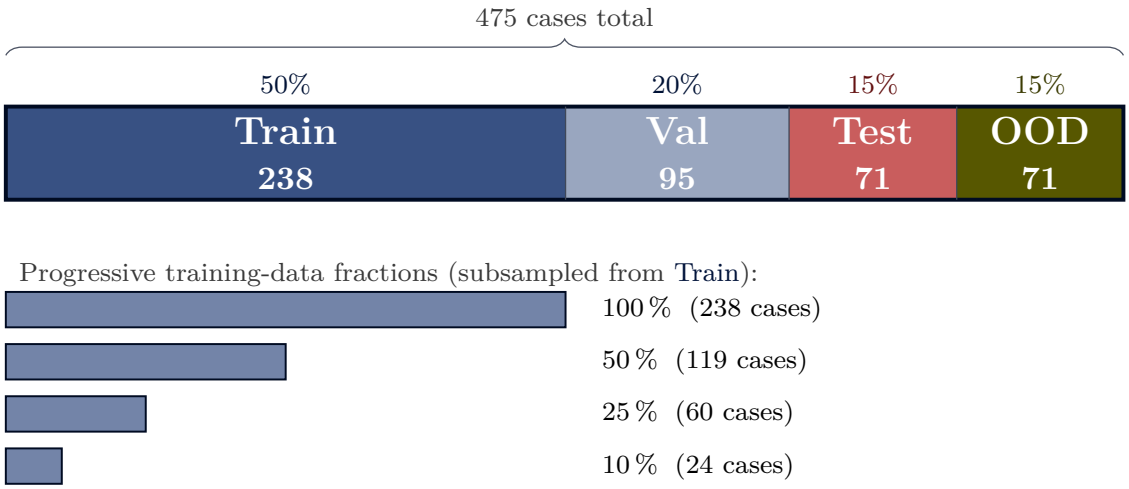

Figure 3.4: Data split for the thermal-field dataset (475 cases): the train/validation/in-distribution-test/out-of-distribution (OOD) partition (Train 238, Val 95, Test 71, OOD 71) and the progressive training-data fractions (10/25/50/100%) subsampled from the training set for the data-efficiency study.

4

Results and Discussion

4.1 Benchmark Setup

In this study we set out to evaluate six models: FNO and PINO (spectral operators), DeepONet and DeepONet+PDE (branch-trunk networks), and Point MLP and pPINN (pointwise MLPs). We train each model across four training-data fractions (10%, 25%, 50%, 100%) and three seeds resulting in a total of 72 completed runs upon which this analysis is based. All the models share the exact same pipeline for training as described in Chapter 3.

Table 4.1: Models used in final benchmark.

Model	Family	Role	Physics	Parameters
FNO	Spectral operator	Data only	No	260,661
PINO	Spectral operator	PDE regularised	Yes	260,661
DeepONet	Branch-trunk	Data only	No	25,601
DeepONet+PDE	Branch-trunk	PDE regularised	Yes	25,601
Point MLP	Pointwise coordinate	Data only	No	20,161
pPINN	Pointwise coordinate	PDE regularised	Yes	20,161

4.2 Which Architecture Wins?

We can see from Table 4.2 that Point MLP has the lowest test RMSE at 12.1 ± 0.3 K and is closely followed by pPINN at 12.7 ± 1.5 K. FNO and PINO follow at 16.5 ± 2.2 K and 16.4 ± 1.7 K respectively. DeepONet has the highest test RMSE at 18.2 ± 0.5 K, with DeepONet+PDE performing worst at 20.5 ± 0.8 K (see subsection 3.5.4 for split details). We see a clear separation of the model families here on all metrics besides the OOD RMSE.

As we just noted the results for OOD don't have as clear of a separation. pPINN has the lowest OOD RMSE at 40.9 ± 7.9 K, followed by DeepONet+PDE (45.6 ± 3.0 K) and Point MLP (46.9 ± 16.6 K). FNO and PINO have much higher OOD errors at 65.8 ± 23.5 K and 65.0 ± 21.0 K. The data-only DeepONet is far worse at 103.3 ± 43.2 K, with the large variance coming from one seed that produced very poor predictions.

In Table 4.3 and Figure 4.2 we can see additional metrics such as structural similarity (SSIM), melt-pool IoU, peak absolute error and inference latency. Point MLP performs best on SSIM (0.728) and has the lowest peak absolute error (38.9 K)

Table 4.2: Performance on 100% of the data. Values are mean ± standard deviation in Kelvin.

Model	Valid RMSE	Test RMSE	OOD RMSE	Test MAE
FNO	13.5 ± 1.1	16.5 ± 2.2	65.8 ± 23.5	4.8 ± 0.3
PINO	14.0 ± 0.7	16.4 ± 1.7	65.0 ± 21.0	4.8 ± 0.4
DeepONet	17.4 ± 0.2	18.2 ± 0.5	103.3 ± 43.2	5.4 ± 0.3
DeepONet+PDE	20.0 ± 0.9	20.5 ± 0.8	45.6 ± 3.0	6.1 ± 0.2
Point MLP	12.2 ± 0.4	12.1 ± 0.3	46.9 ± 16.6	3.7 ± 0.2
pPINN	12.2 ± 0.2	12.7 ± 1.5	40.9 ± 7.9	4.0 ± 0.6

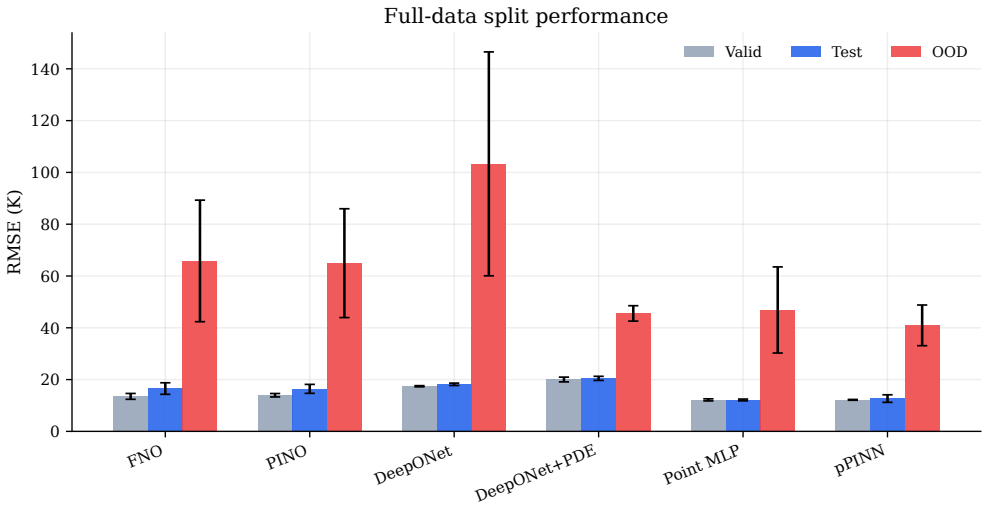

Figure 4.1: Full-data validation, test and process-extreme OOD RMSE. Bars show the mean over all the seeds and error bars show one standard deviation.

while DeepONet shows the highest peak errors (55.4 K) and DeepONet+PDE has the lowest melt-pool IoU (0.601). Again the DeepONet family underperforms in comparison to the two other families.

Table 4.3: Extended random-test diagnostics at the largest available training fraction. Values are mean ± standard deviation over the seeds.

Model	Hot RMSE	Peak abs. error	SSIM	Melt IoU	Inference (ms)
FNO	41.9 ± 2.4	43.6 ± 6.6	0.662 ± 0.026	0.656 ± 0.027	13.89 ± 1.07
PINO	39.4 ± 2.2	40.7 ± 4.3	0.621 ± 0.003	0.683 ± 0.037	13.72 ± 0.60
DeepONet	47.6 ± 1.7	55.4 ± 1.8	0.678 ± 0.014	0.665 ± 0.028	6.97 ± 0.32
DeepONet+PDE	55.6 ± 4.2	53.4 ± 6.7	0.659 ± 0.016	0.601 ± 0.042	6.77 ± 0.10
Point MLP	35.7 ± 1.4	38.9 ± 1.6	0.728 ± 0.000	0.673 ± 0.050	8.63 ± 0.23
pPINN	37.4 ± 3.6	42.9 ± 2.6	0.723 ± 0.005	0.681 ± 0.017	8.15 ± 0.29

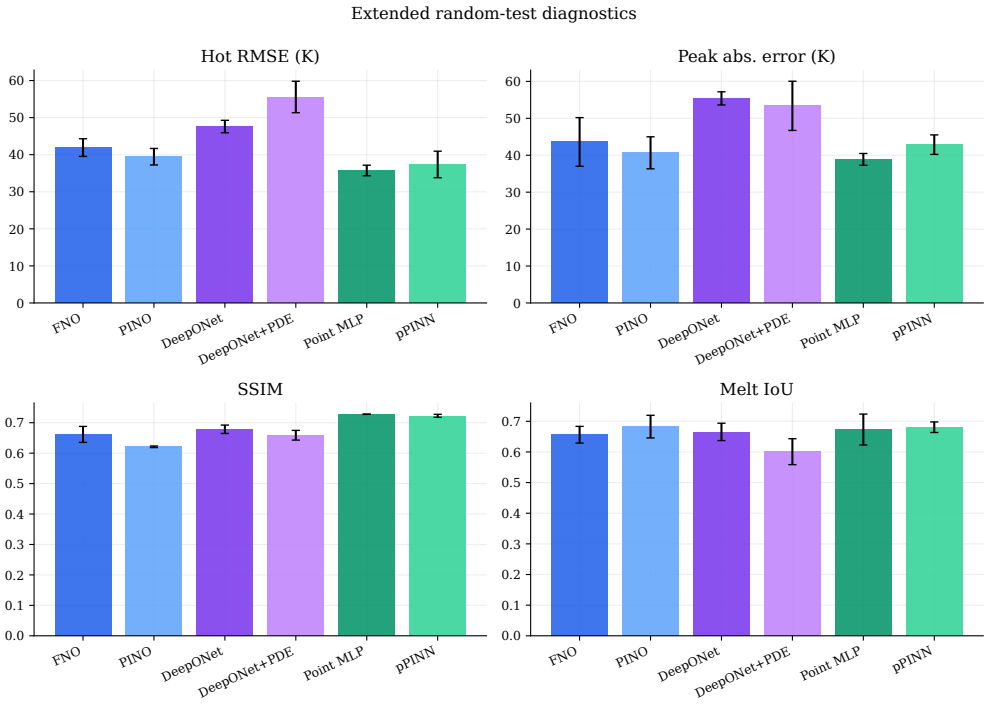

Figure 4.2: Extended diagnostics on the test split at full training data. Top left shows hot-zone RMSE, top right is peak absolute error, bottom left is structural similarity (SSIM) and melt-pool IoU is in the bottom right panel. Point MLP and pPINN dominate on all the accuracy metrics.

Figure 4.3 shows a comparison of all six models on the easy test case `run_402`, which has a small melt pool (peak temperature 585 K). Point MLP has the lowest slice-level RMSE (2.5 K), but the differences between models are subtle. To expose the differences more clearly, Figure 4.4 shows a second case `run_142` with a normal melt pool (peak temperature 1909 K), where the gaps between architectures are much more pronounced.
Given the setup of this problem the results do make sense. The input space, i.e. the process parameters, is low-dimensional and the output space is fixed. We don't use

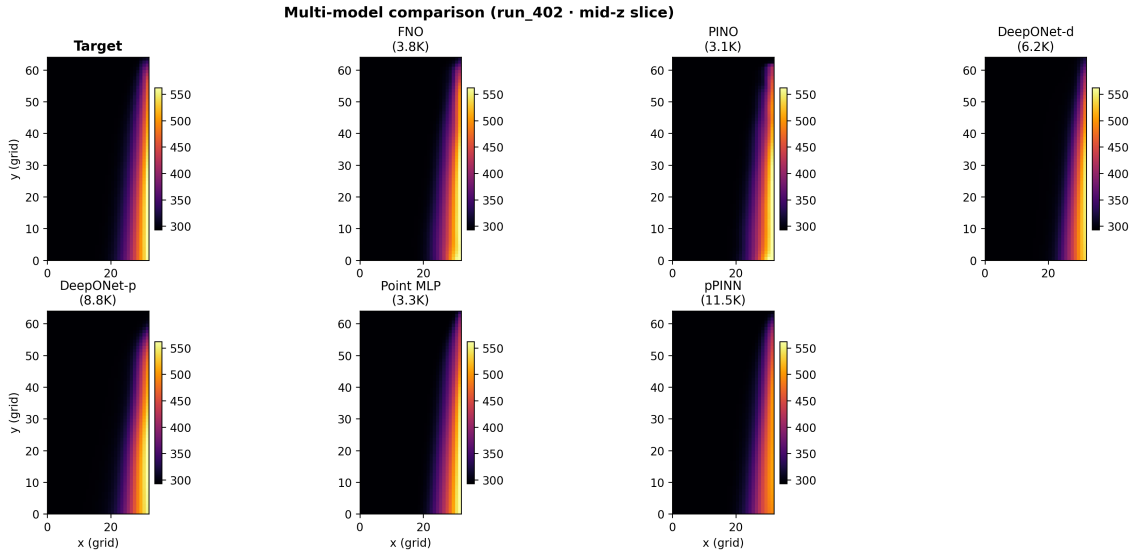

Figure 4.3: Comparison for the easy test case `run_402` (peak 585 K) across all models. Top-left panel contains the FVM simulation. The remaining panels show predictions from each model family. Slice-level RMSE is shown beneath each model label.

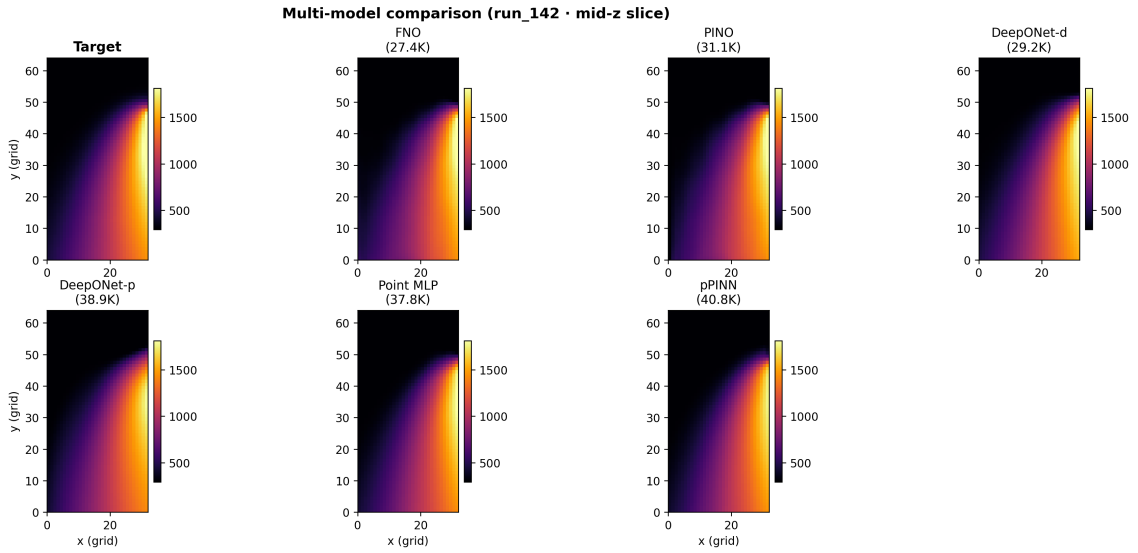

Figure 4.4: Comparison for the normal test case `run_142` (peak 1909 K) across all models. Top-left panel contains the FVM simulation. The remaining panels show predictions from each model family. Slice-level RMSE is shown beneath each model label.

any material-dependent properties which ends up hurting DeepONet performance, since we end up with a parametric regression task which does not utilise its full potential. DeepONet is designed to encode an input field at fixed sensor locations but we are just giving it four scaled process parameters.
Table 4.4 shows the worst OOD cases across all models. DeepONet fails entirely here, with zero IoU on the worst case, which is consistent with its high OOD RMSE. The physics-informed variants do hold up better but the errors are still large.
Across all in-distribution metrics, Point MLP is the clear winner, with the lowest test RMSE, best SSIM, lowest hot-zone RMSE and lowest peak absolute error. The physics-informed variants perform similarly to their data-only counterparts. A notable outlier is the physics-informed DeepONet on OOD RMSE, which offers significant improvements over its data-only variant. This result will be further analysed in Section 4.4.

Table 4.4: Worst OOD cases by peak-temperature absolute error in the extended per-case diagnostics.

Model	Split	Case	RMSE	Peak abs.	Hot RMSE	Melt IoU
DeepONet	ood	run_63	797.2	1160.5	1496.9	0.000
DeepONet	ood	run_438	345.1	1113.1	622.0	0.041
DeepONet	ood	run_438	371.0	1111.6	589.5	0.055
Point MLP	ood	run_438	177.8	949.3	410.5	0.574
pPINN	ood	run_438	93.4	913.0	174.3	0.709
pPINN	ood	run_438	137.0	902.3	257.2	0.681
pPINN	ood	run_438	123.0	902.3	232.6	0.479
Point MLP	ood	run_438	118.1	886.7	289.6	0.634
DeepONet	ood	run_63	715.4	860.3	1170.3	0.007
DeepONet	ood	run_438	86.4	850.5	154.5	0.687

4.3 How Much Data Is Enough?

To evaluate data efficiency we trained each model using 10%, 25%, 50% and 100% of the training data. All models improve significantly between 10% and 25% data. Point MLP performs the best at 25% data (18.3 ± 0.6 K) and remains the best model at 50% (14.9 ± 0.9 K) and 100% (12.1 ± 0.3 K) on the test split, as shown in Table 4.5. The physics term does not give a clear advantage at any data fraction, and at 100% data the physics-informed variants perform no better than their data-only counterparts on the test split. This is worth noting since data efficiency in low-data regimes is one of the main motivations for using physics-informed training.
From Table 4.6 and Figure 4.6 we can see that the optimal training fractions for OOD performance varies depending on the model family. For the pointwise MLP family OOD RMSE improves monotonically with more data, reaching 46.9 ± 16.6 K and 40.9 ± 7.9 K for Point MLP and pPINN respectively at 100%. In contrast FNO and PINO show their best OOD at the intermediate fractions (50% and 25% respectively)

Table 4.5: Random-test RMSE across training data fractions. Values are mean ± standard deviation over the seeds in Kelvin.

Model	**10%**	**25%**	**50%**	**100%**
FNO	34.6 ± 4.2	20.0 ± 1.0	16.9 ± 1.6	16.5 ± 2.2
PINO	35.4 ± 2.8	20.0 ± 0.7	17.4 ± 2.2	16.4 ± 1.7
DeepONet	37.9 ± 6.7	22.5 ± 2.5	19.8 ± 0.9	18.2 ± 0.5
DeepONet+PDE	38.5 ± 7.6	25.6 ± 1.9	21.7 ± 3.2	20.5 ± 0.8
Point MLP	33.1 ± 4.0	18.3 ± 0.6	14.9 ± 0.9	12.1 ± 0.3
pPINN	35.0 ± 6.8	18.4 ± 2.1	14.5 ± 2.2	12.7 ± 1.5

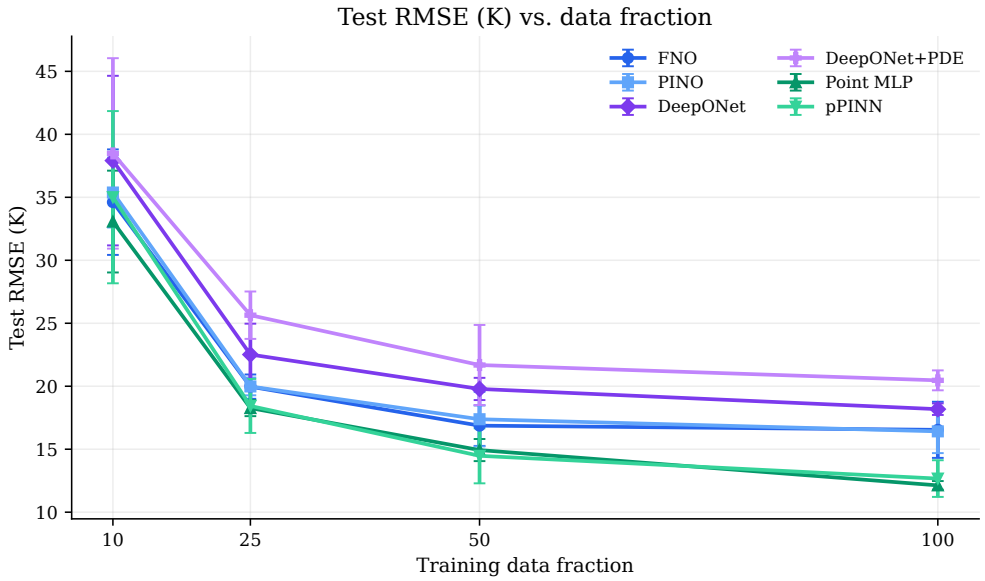

Figure 4.5: Test RMSE versus training-data fraction. All models improve sharply from 10% to 25% data. The pointwise MLP family gives the strongest test performance at the higher data fractions.

and get worse at 100% data. The data-only DeepONet follows a similar pattern.

Table 4.6: Process-extreme OOD RMSE across training data fractions. Values are mean ± standard deviation seeds in Kelvin.

Model	10%	25%	50%	100%
FNO	86.3 ± 16.3	53.5 ± 7.6	51.5 ± 6.1	65.8 ± 23.5
PINO	86.9 ± 9.4	52.4 ± 7.5	52.5 ± 5.7	65.0 ± 21.0
DeepONet	91.1 ± 15.1	53.9 ± 7.2	46.6 ± 3.2	103.3 ± 43.2
DeepONet+PDE	86.8 ± 12.5	66.7 ± 6.3	97.0 ± 37.1	45.6 ± 3.0
Point MLP	84.6 ± 14.6	41.7 ± 0.6	47.0 ± 5.8	46.9 ± 16.6
pPINN	82.5 ± 23.4	45.3 ± 4.9	42.5 ± 4.6	40.9 ± 7.9

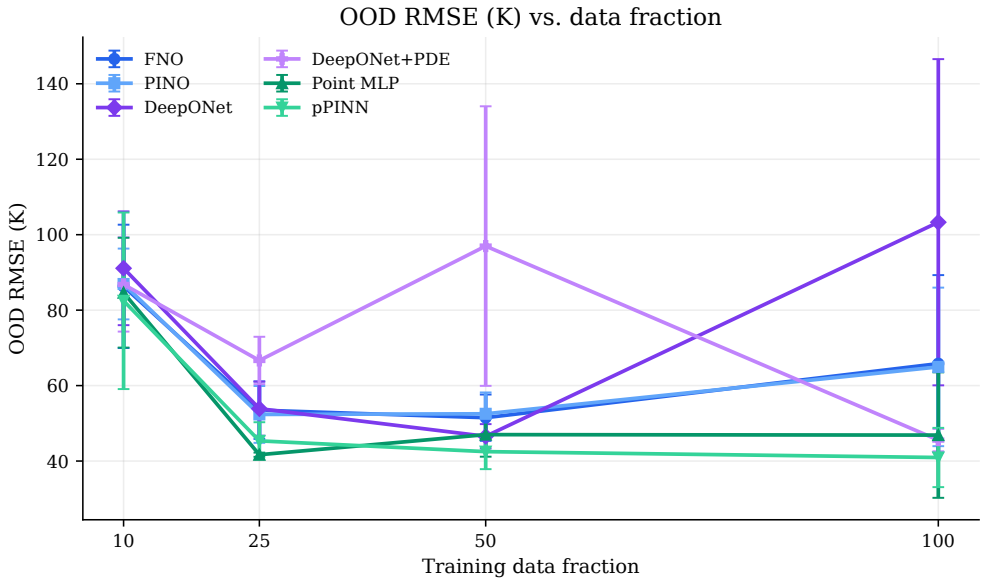

Figure 4.6: Process-extreme OOD RMSE versus training-data fraction for all six models. Each line is one model; error bars show one standard deviation across seeds. Point MLP and pPINN improve monotonically with more data, whereas FNO, PINO and DeepONet show their best OOD performance at intermediate fractions and worsen at 100%, suggesting overfitting to the interior of the training distribution.

The poor performance of FNO/PINO when training on 100% of the data could be explained by overfitting to the interior of the process parameter range since early stopping is driven by validation data which is in the interior of the range. This means that training on a smaller dataset for these models acts as a sort of implicit regulariser. Based on this we may choose to include extreme OOD cases in the validation split to avoid this.
In summary, increasing the training data fraction consistently improves in-distribution performance for all models, with Point MLP benefiting the most. For OOD generalisation, the picture is more nuanced: the pointwise MLP models improve monotonically with more data, whereas the spectral operators (FNO/PINO) actually generalise

better at intermediate data fractions, suggesting that the larger datasets cause them to overfit to the interior of the training distribution.

4.4 Does Physics-Informed Training Help?

Since we designed a paired ablation design we can easily evaluate whether adding the physics residual improved performance. Table 4.7 and Figure 4.7 show physics-minus-data RMSE deltas over matched seeds (i.e., comparing physics-informed and data-only variants that were initialised with the same random seed) where negative values mean that the physics-informed variant is better than its data-only counterpart.

Table 4.7: Physics-minus-data RMSE deltas over matched seeds. Negative values indicate that the physics-informed variant improved over its data-only counterpart.

Architecture	Data	Δ Test RMSE	Δ OOD RMSE
FNO	10%	0.8 ± 1.7	0.6 ± 7.1
FNO	25%	0.0 ± 0.6	-1.1 ± 1.9
FNO	50%	0.5 ± 1.0	1.0 ± 1.1
FNO	100%	-0.1 ± 0.8	-0.9 ± 4.1
DeepONet	10%	0.6 ± 1.0	-4.3 ± 2.6
DeepONet	25%	3.1 ± 1.1	12.8 ± 13.2
DeepONet	50%	1.9 ± 2.9	50.4 ± 36.3
DeepONet	100%	2.3 ± 1.1	-57.7 ± 44.5
Point MLP	10%	1.9 ± 3.0	-2.1 ± 8.8
Point MLP	25%	0.2 ± 2.4	3.7 ± 4.5
Point MLP	50%	-0.5 ± 1.5	-4.5 ± 9.2
Point MLP	100%	0.5 ± 1.7	-5.9 ± 12.5

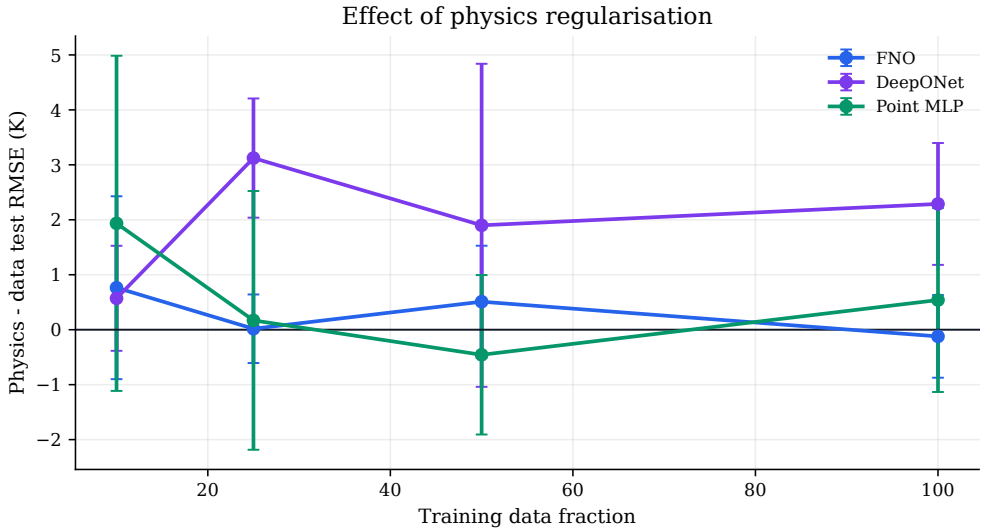

Figure 4.7: Effect of physics regularisation on test RMSE. Positive values indicate that the physics-enabled variant has higher error than the matched data-only variant.

The results clearly show that the physics-informed variants do not improve volume-averaged temperature prediction for the in-distribution data. The OOD deltas are mixed with some showing improved performance and others suffering. There is no clear pattern or evidence of physics-informed models outperforming their data-driven counterparts.
Figure 4.8 shows the individual loss terms during training of the pPINN model at full data. The composite data loss decreases by orders of magnitude during training, from roughly 10^{-2} to 10^{-4}. The raw PDE residual remains large (roughly 10^{1}) and never converges, which confirms that the simplified heat equation is not satisfied by the reference FVM simulation fields in our model. The adaptive weighting scheme keeps the weighted PDE contribution to approximately 2% of the total loss and the boundary term contributes roughly 5%. The physics terms together contribute less than 7%, and we can see that supervised data dominates.

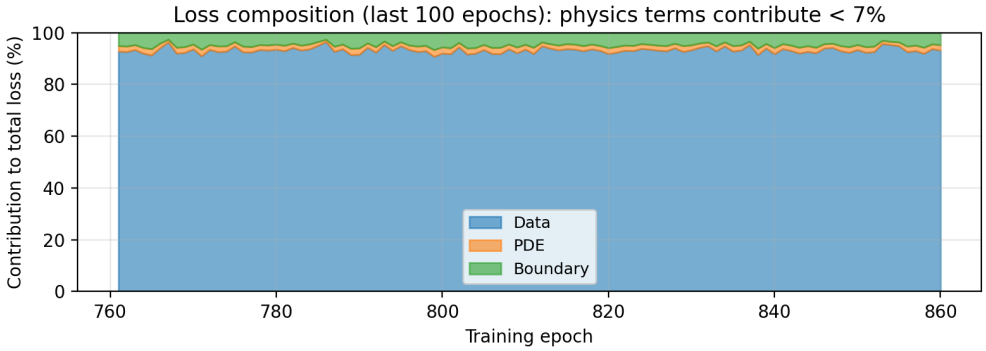

Figure 4.8: Loss composition during pPINN training at full data (seed 132). The stacked area shows the fraction of the total weighted loss contributed by the data term (blue), the PDE residual (orange) and the boundary condition (green) during the last 100 epochs.

The supervised thermal fields already provide a strong learning signal. The data-only models have enough information to learn the in-distribution predictions accurately, and the loss composition confirms this.
Leaving out the heat source term Q limits the usefulness of the physics signal. As noted in Section 3.3, the FVM data include laser-bead interaction that the fixed-source quasi-steady-state model cannot capture. For example, at high feed rates a large bead builds up and intercepts the laser, altering the energy deposition on the substrate, so modelling Q precisely is not possible from the substrate field alone. The residual evaluated is therefore $\mathcal{N}[T] = \rho(T)\, c_p(T)\, v\, \partial T/\partial \xi + \nabla\cdot(k(T)\nabla T)$, the source-free part of the operator. Near the heat source forcing $\mathcal{N}[\hat{T}_\theta] \approx 0$ conflicts with the true physics ($\mathcal{N}[T] = -Q \neq 0$), while away from the source $Q \approx 0$ and the residual is approximately correct.
The reference simulation fields are not exact zero-residual samples under even the full PDE operator either. Forcing the neural surrogate towards this residual might hurt performance due to the mismatch.
Including the heat source term in the residual and improving the overall fidelity of the physics operator would probably improve the usefulness of the physics con-

straint. However, as shown in Section 4.7, the physics-informed variants do provide regularisation benefits at high temperatures on OOD cases.

4.5 What Is the Cost-Accuracy Trade-off?

Table 4.8 shows the model size and training time at 100% data. The physics-informed variants are all much slower than their data-driven counterparts without improving predictive accuracy. Therefore the current physics terms are not cost-effective in our setup.

Table 4.8: Full-data training cost and random-test accuracy over seeds. One outlier seed per model excluded from Point MLP and pPINN training-time statistics (see footnote). Best (lowest) per column in bold.

Model	Parameters	Training time (s)	Best epoch	Test RMSE (K)
FNO	260,661	9365 ± 3302	343 ± 65	16.5 ± 2.2
PINO	260,661	13602 ± 763	330 ± 46	16.4 ± 1.7
DeepONet	25,601	6361 ± 482	650 ± 242	18.2 ± 0.5
DeepONet+PDE	25,601	7091 ± 657	220 ± 66	20.5 ± 0.8
Point MLP	20,161	8773 ± 967†	440 ± 130	**12.1 ± 0.3**
pPINN	20,161	11912 ± 785†	455 ± 95	12.7 ± 1.5

† One seed excluded per model due to abnormally long wall time caused by system sleep during training.

Point MLP has the best combination of accuracy and parameter count, with only 20 161 trainable parameters. Its training time is comparable to the spectral operators (FNO/PINO). DeepONet trains fastest at roughly 6 400 s.
The fastest inference is produced by the DeepONet family at roughly 7 ms per prediction, and closely followed by Point MLP and pPINN at 8-9 ms. The KATLA FVM solver takes approximately one minute per simulation on GPU, which means that Point MLP offers a speed-up of roughly 7 000×. This makes the surrogate practical for interactive process parameter exploration where hundreds of candidate conditions need to be evaluated rapidly. Figure 4.9 summarises the training cost versus test accuracy trade-off, while Figure 4.10 shows the inference latency versus test accuracy. Point MLP achieves the best combination of accuracy and inference speed.
This speed-up naturally suggests a practical two-stage workflow where we could use the surrogate to screen hundreds of candidate parameter combinations interactively at milliseconds per prediction, then run FVM simulations on the top candidates for final validation. This keeps the physical fidelity of the solver where it matters most while avoiding the need to simulate unproductive regions of the design space.

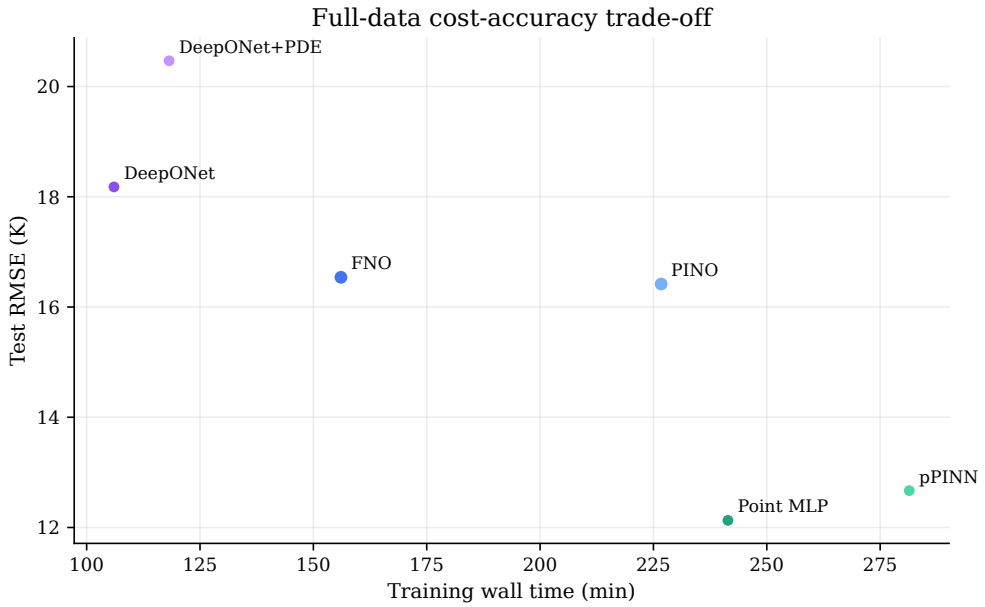

Figure 4.9: Training cost versus test RMSE at full data. Smaller markers indicate fewer parameters. The physics variants require longer training time but do not improve accuracy.

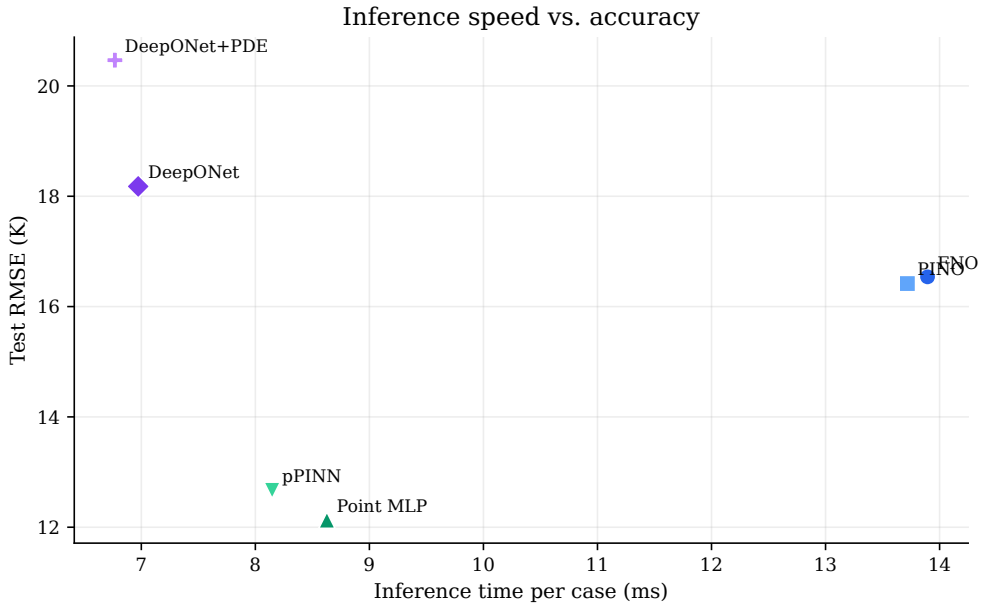

Figure 4.10: Inference latency versus test RMSE. Point MLP achieves the best accuracy at the lowest inference cost. FNO and PINO are roughly twice as slow without giving any accuracy advantage.

4.6 Where Do Models Fail?

The best test RMSE is about 12 K while the best OOD RMSE is around 41-47 K. This gap, which is generally consistent across models, confirms that the surrogate interpolates well within the training envelope but struggles at process parameter extremes.
On OOD cases the main failure mode is systematic under-prediction of peak temperature and melt-pool extent. The worst OOD case (`run_63`) has a large melt pool and the model under-predicts both the peak temperature and the melt-pool extent, giving 190.8 K RMSE. By contrast the best test case (`run_402`) has such a small melt pool that it is barely visible and most of the domain is ambient. The prediction is accurate with 2.5 K RMSE, but this reflects the fact that there is not much thermal structure to get wrong. Figures 4.11 and 4.12 show both cases. The liquidus contour overlay in Figure 4.14 backs this up. On the worst OOD case the predicted melt pool is visibly smaller than the FVM simulation.

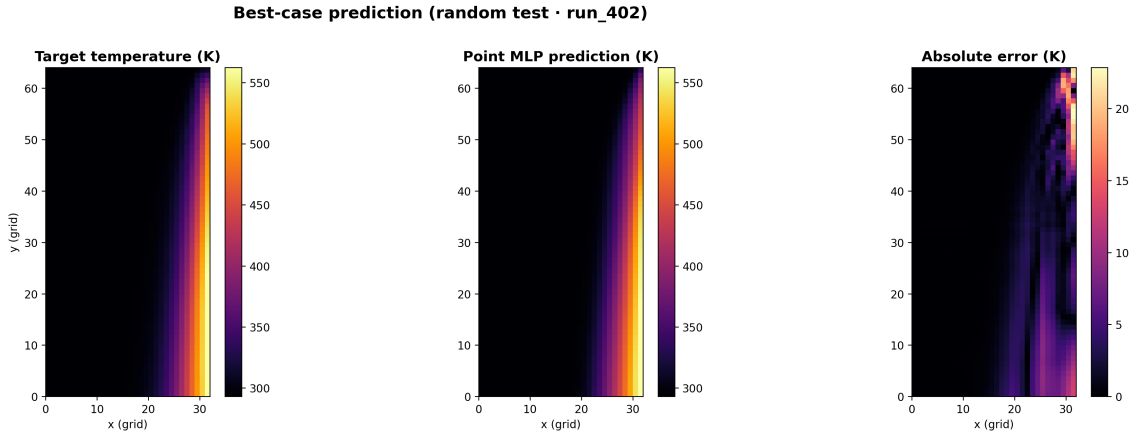

Figure 4.11: Best-case test case (`run_402`). Centre-plane temperature slices for FVM simulation (left), Point MLP prediction (centre) and absolute error (right). The prediction matches the FVM simulation in peak temperature and melt-pool extent. The RMSE for this case is 2.5 K.

In Figure 4.13 the per case RMSE is mapped onto the process parameter space (laser power versus scan speed) for Point MLP. The errors grow monotonically with melt-pool size, with the least accurate cases clustering at very low scan speeds. These cases are difficult primarily because the training data is sparse in this region; the underlying physics is not inherently harder to approximate here.
This means predictions at very low scan speeds should be treated with caution, and a practical workflow should flag candidate parameters near the edge of the training envelope. The observation that error grows with melt-pool size also connects to the next section since the volume of a case is dominated by ambient voxels, the aggregate RMSE is most sensitive to ambient-region prediction, while the actual error is in the thermally active region. This ambient bias in the metric explains why the best-case prediction (small melt pool, mostly ambient) looks so good compared to the worst case.

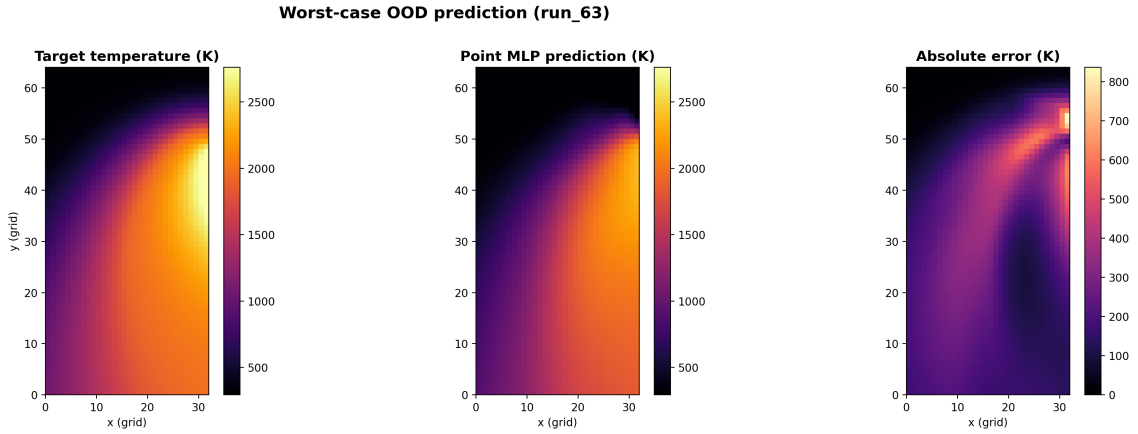

Figure 4.12: The worst case process-extreme OOD case (`run_63`). Centre-plane temperature slices for FVM simulation (left), Point MLP prediction (centre) and absolute error (right). The peak temperature is under-predicted and the melt-pool region is narrower than the FVM simulation. The RMSE for this case is 190.8 K for Point MLP.

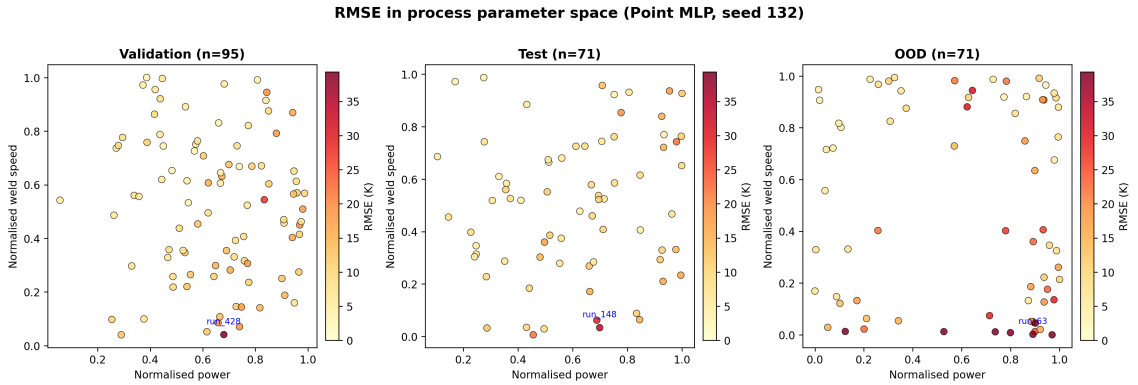

Figure 4.13: Per case RMSE for Point MLP (seed 132, 100% data) mapped onto the normalised power-scan-speed plane (shown as "weld speed" on the axis), shown separately for validation, test and OOD splits. Axes show parameters scaled by the model's input normalisation (hiding real process values); the colour bar shows RMSE in Kelvin. Each point is a single case, coloured by its RMSE (yellow = low error, red = high error). The worst case in each split is annotated in blue.

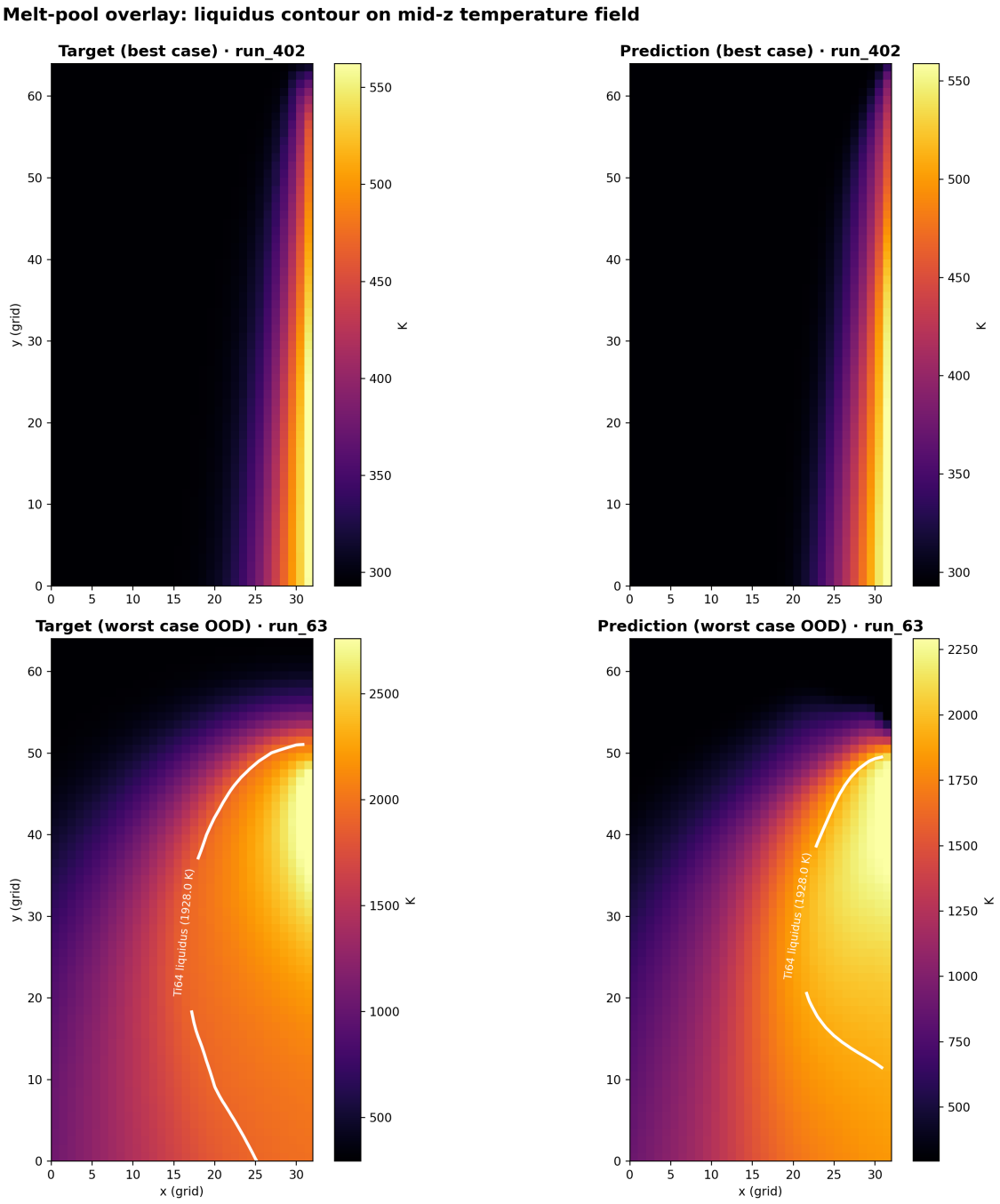

Figure 4.14: Liquidus-threshold melt-pool contours (1928 K) overlaid on the FVM simulation thermal field. Top row shows the best-case test instance (`run_402`) with good agreement between target and prediction. Bottom row shows the worst OOD instance (`run_63`) where the predicted melt pool is much smaller than the FVM simulation. The white contour line marks the liquidus isotherm.

4.7 Ambient Bias in Volume-Averaged Metrics

The aggregate RMSE used throughout this thesis is averaged across all the voxels. In our benchmark approximately 87% of voxels in the test split are at ambient temperature (<400 K) while the thermally active region (≥800 K) occupies less than 7% of the volume. Table 4.9 and Figure 4.15 show per-bin RMSE across nine temperature intervals for all models at 100% data in order to give us a more nuanced view of the performance of the models.

Table 4.9: Temperature-stratified RMSE at the largest training fraction, showing how prediction error varies across temperature bins. Values are mean ± standard deviation over seeds. Bins are defined by the target (ground-truth) temperature. The ambient bin (<400 K) dominates the volume-averaged RMSE.

Random-Test									
Model	293-400	400-500	500-600	600-800	800-1 000	1 000-1 200	1 200-1 500	1 500-1 928	≥ 1 928
	RMSE (K) per target-temperature bin								
FNO	6.2 ± 0.8	28.9 ± 3.0	34.3 ± 4.5	40.8 ± 5.4	47.9 ± 8.5	57.3 ± 12.7	60.9 ± 13.9	54.1 ± 5.7	51.4 ± 9.5
PINO	7.0 ± 0.5	32.0 ± 2.6	35.9 ± 3.6	39.6 ± 3.9	45.3 ± 5.8	52.6 ± 8.9	57.7 ± 11.0	49.9 ± 6.0	48.0 ± 10.3
DeepONet	9.6 ± 1.2	29.6 ± 4.0	34.2 ± 3.3	39.3 ± 1.5	42.5 ± 5.6	49.0 ± 7.8	58.3 ± 10.4	75.9 ± 14.8	93.5 ± 9.8
DeepONet+PDE	13.0 ± 0.5	36.6 ± 1.4	38.6 ± 1.9	40.7 ± 3.1	42.4 ± 3.5	45.0 ± 4.1	52.3 ± 3.6	78.5 ± 10.1	115.0 ± 13.2
Point MLP	5.5 ± 0.2	20.4 ± 2.8	24.2 ± 2.4	28.7 ± 1.3	33.2 ± 1.9	38.2 ± 0.7	39.6 ± 0.1	45.8 ± 7.3	51.6 ± 15.4
pPINN	5.2 ± 0.6	22.5 ± 4.5	27.6 ± 5.1	31.8 ± 5.0	35.8 ± 5.1	42.5 ± 4.7	42.6 ± 1.9	40.6 ± 3.1	43.9 ± 2.3
OOD									
Model	293-400	400-500	500-600	600-800	800-1 000	1 000-1 200	1 200-1 500	1 500-1 928	≥ 1 928
	RMSE (K) per target-temperature bin								
FNO	12.1 ± 0.8	50.4 ± 9.1	76.0 ± 20.5	112.5 ± 37.2	156.9 ± 62.0	194.4 ± 87.6	233.2 ± 100.7	239.4 ± 77.4	199.1 ± 56.0
PINO	12.6 ± 0.3	54.8 ± 5.4	82.2 ± 12.8	115.6 ± 25.0	152.5 ± 48.4	185.3 ± 74.9	226.0 ± 98.3	235.5 ± 84.5	194.4 ± 57.3
DeepONet	20.6 ± 5.8	59.2 ± 9.3	88.8 ± 25.0	132.3 ± 51.6	190.6 ± 82.5	253.1 ± 111.4	335.2 ± 148.0	441.7 ± 191.8	544.3 ± 266.1
DeepONet+PDE	19.9 ± 0.6	62.8 ± 2.6	83.0 ± 5.9	96.3 ± 10.5	101.7 ± 12.1	103.4 ± 11.6	109.5 ± 8.4	124.6 ± 11.1	136.9 ± 18.9
Point MLP	11.2 ± 0.2	31.6 ± 4.7	42.7 ± 6.6	62.9 ± 14.9	94.4 ± 29.8	133.6 ± 59.9	162.2 ± 64.1	181.7 ± 63.7	210.2 ± 97.8
pPINN	10.7 ± 0.7	32.7 ± 3.2	44.8 ± 6.4	60.1 ± 7.9	79.3 ± 7.8	103.1 ± 13.2	143.0 ± 35.0	166.9 ± 49.6	162.0 ± 49.4

Two patterns become apparent in the thermally active region. Firstly, RMSE increases monotonically with target temperature for all models, but the rate of growth differs a lot between them. Point MLP and pPINN show the gentlest slope, reaching only 41-46 K RMSE at the 1500-1928 K bin, while DeepONet reaches 76 K and DeepONet+PDE reaches 79 K at the same bin. Interestingly, FNO/PINO also become more competitive at the higher temperatures and rival the performance of the Point MLP and pPINN models. This observation does not hold for the OOD stratification.

The OOD stratification reveals the physics regularisation that is not visible in the aggregate metric. On the OOD split DeepONet+PDE maintains a flat error curve from 800 K upward, ranging from 102 to 137 K across all high-temperature bins, compared to data-only DeepONet's steep climb from 132 K to 544 K. Similarly pPINN outperforms Point MLP in the 800-1200 K OOD range. The physics term provides strong regularisation at high temperatures on OOD cases, preventing the poor extrapolation seen in the data-only variant. This is confirmed by the within-family RMSE ratio in Figure 4.16, where DeepONet+PDE/DeepONet drops well below 1.0 at high temperatures on the OOD split. This aligns with our analysis from Section 4.4 where we note that the source-free PDE residual is approximately correct and this gives a stabilising effect for extrapolation even when it slightly hurts test accuracy.

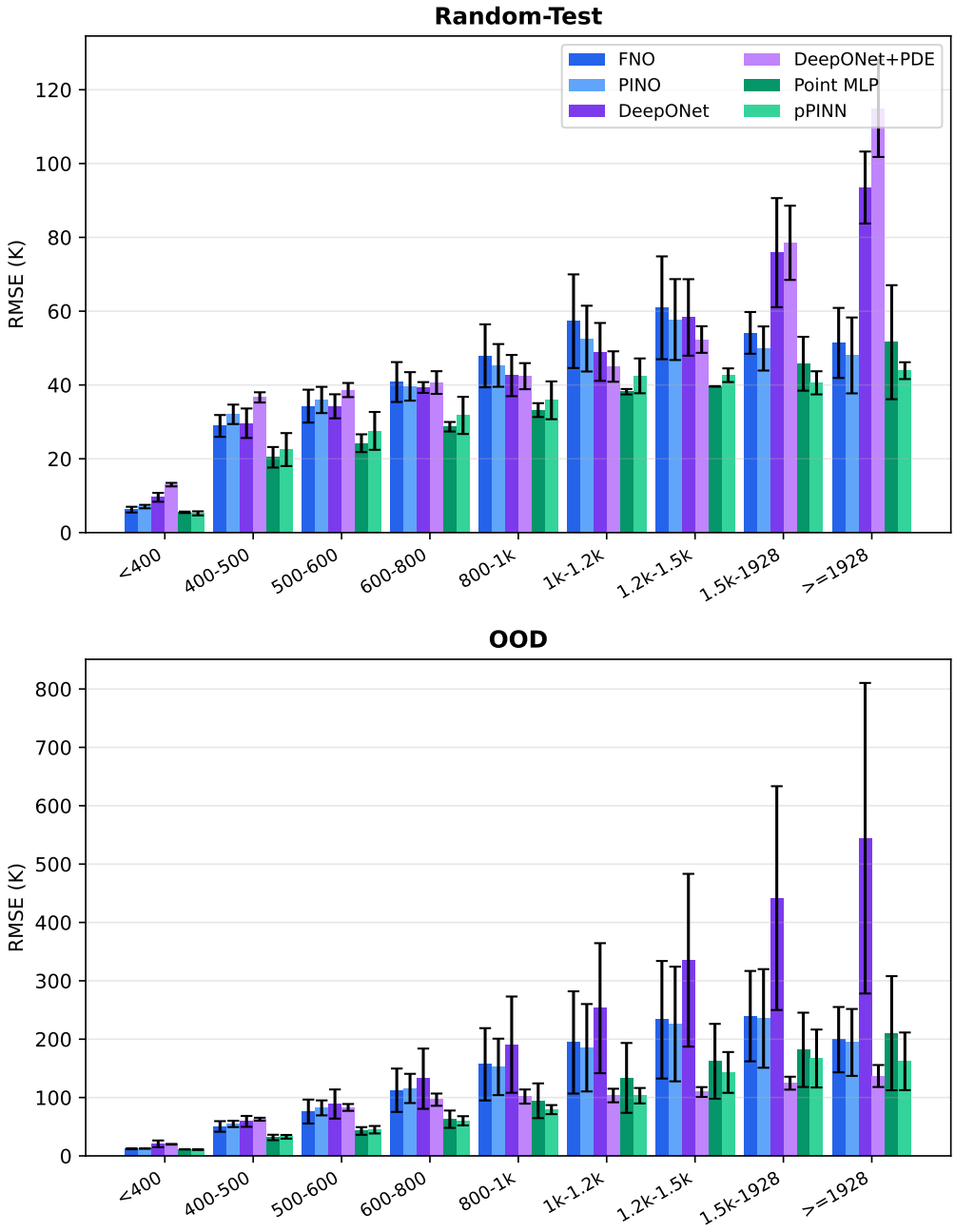

Figure 4.15: Temperature-stratified RMSE for all six models on the test and OOD splits at 100% data. Error bars show standard deviation across seeds. The ambient bin (<400 K) dominates the volume-averaged metric; in the thermally active region (≥800 K), pointwise MLP models maintain lower error and slower error growth with temperature than the spectral operators.

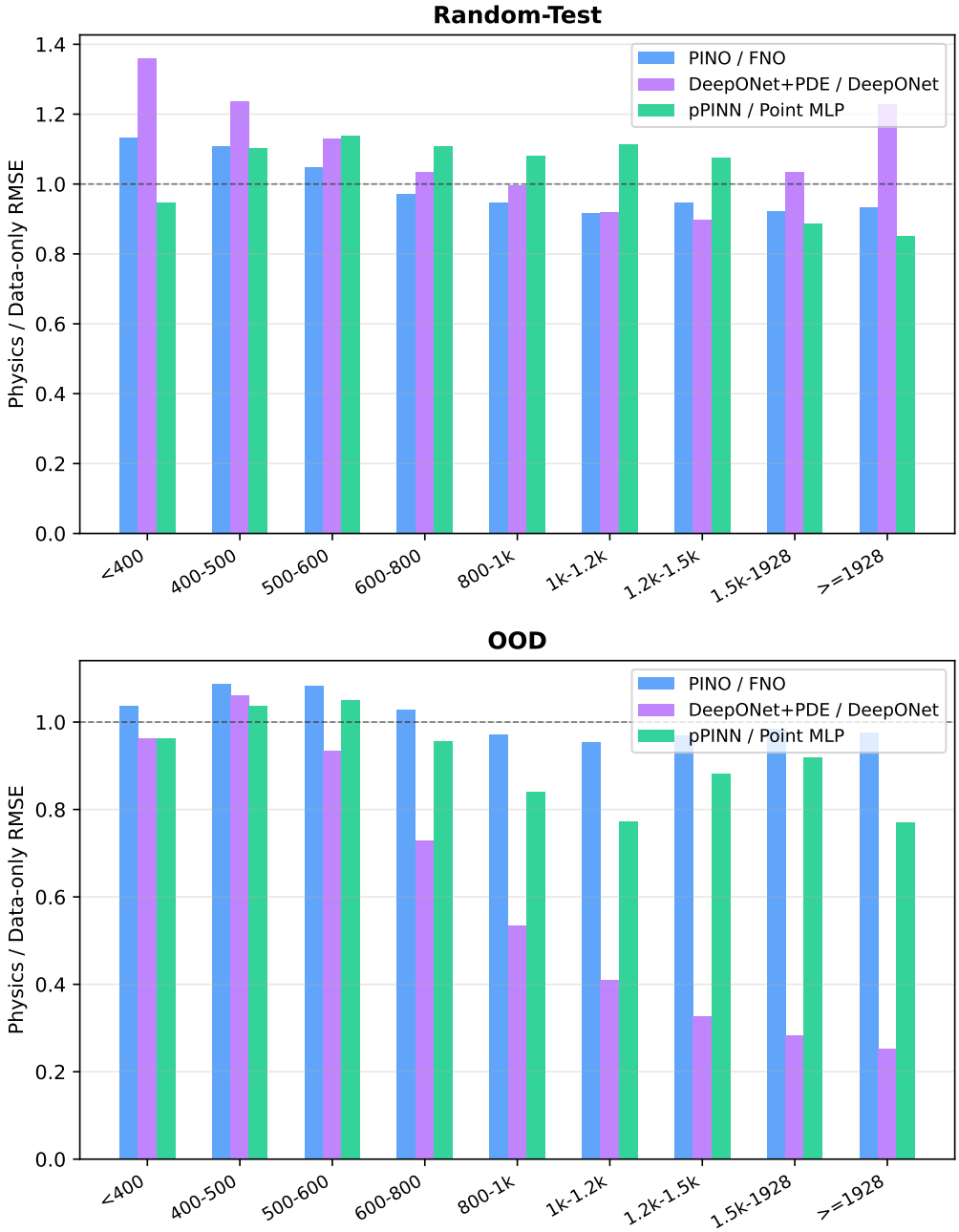

Figure 4.16: Within-family physics/data RMSE ratio for each model family on the test and OOD splits at 100% data. A value of 1.0 means the physics-informed and data-only variants perform equally; values above 1.0 mean the physics variant is worse. The dashed line marks equal performance. PINO/FNO is consistently above 1.0 at high temperatures, while DeepONet+PDE/DeepONet drops well below 1.0 on the OOD split.

The claim that physics-informed learning offers no benefit becomes weaker. Despite the lack of performance on the test split, the physics loss generally improves OOD performance in the high-temperature region. This doesn't change the overall recommendation though since the ambient-dominated volume overwhelms the aggregate metric and the data-only model is still the better choice for this task.

4.8 Summary of Model Performance

The **pointwise MLP family** (Point MLP, pPINN) delivers the best overall in-distribution accuracy and has a fast inference speed, lagging slightly behind the DeepONet family. Point MLP achieves the lowest test RMSE with the fewest trainable parameters. The main limitation is the lengthier training time required by this family. We note that the physics-informed variant (pPINN) does not improve aggregate accuracy but provides regularisation at high temperatures on OOD cases.
The **spectral operator family** underperforms compared to the pointwise models on most metrics. The physics-informed variant (PINO) provides no clear advantage over FNO.
The **branch-trunk operator family** (DeepONet, DeepONet+PDE) achieves the fastest inference times but shows the weakest accuracy. The parametric variant used here does not fully exploit the branch-trunk architecture's potential, since the branch receives only four scalar process parameters rather than a spatially varying input function. DeepONet+PDE provides strong OOD regularisation at high temperatures but suffers from the worst in-distribution accuracy.

5
Conclusion

In this thesis we have investigated machine-learning surrogates for DED thermal-field prediction with a focus on whether physics-informed training improves accuracy and OOD generalisation. We ran a paired benchmark over six model variants, four training-data fractions and three seeds to give a total of 72 runs.
The main findings are as follows. A compact data-only Point MLP achieves the best in-distribution accuracy (12.1 ± 0.3 K test RMSE) with only 20 161 parameters, clearly outperforming the spectral operators and DeepONet on this task. This shows that neural operators are not automatically the best choice when the input is low-dimensional, i.e. we only have a few input process parameters. The tested physics penalties do not improve the overall accuracy; the implemented PDE residual leaves out the heat source term Q (since the evolving bead geometry above the substrate makes the energy deposition profile unknown) and contributes less than 7% of the total weighted loss, so the supervised data dominates. The temperature-stratified analysis does however reveal that physics-informed variants provide regularisation at high temperatures on OOD cases, but this benefit is hidden by the ambient-dominated aggregate metric. Process-extreme OOD generalisation remains the dominant limitation, with OOD RMSE growing monotonically with melt-pool size irrespective of the model.
The main contributions of this work are the following:

1. A reproducible DED thermal-surrogate benchmark with a full shared pipeline for the models.
2. A paired comparison of data-only and physics-regularised variants across three architecture families, demonstrating the lack of aggregate benefit from including physics in the training loss for this problem setting.
3. A demonstration that a compact pointwise surrogate can outperform larger operator models for certain types of problems.
4. A temperature-stratified error analysis that quantifies the ambient bias in the metrics and reveals where models actually differ in the thermally active region.

We recommend the data-only Point MLP as the default surrogate for process-parameter screening based on this benchmark. An RMSE of 12 K is small relative to the liquidus temperature (1928 K) and sufficient for screening process parameters interactively, though it may not be adequate for applications requiring precise melt-pool boundary prediction. The physics-enabled variants are slower and do not improve accuracy. The surrogate should not be trusted solely on test RMSE since high-error OOD cases cluster at very low scan speeds. A practical workflow would be to use the surrogate to screen hundreds of candidates interactively at millisecond

per prediction speed and then run FVM simulations on the top candidates for final validation.
The main limitations of this study are that the training data come from an FVM solver rather than experimental measurements, meaning the surrogate has not been validated against real-world deposition cases. Additionally, only three random seeds were used and a single material and grid resolution was evaluated.
The highest-priority follow-up is to test whether these results generalise beyond the current setup. This benchmark uses a single coarse grid, a small number of process parameters and relatively small models. A meaningful next step is to re-run the benchmark at a finer grid resolution with correspondingly larger models to see if the ranking holds at production-relevant fidelity. Including the heat source term Q in the physics residual is a natural next step for physics-informed training, and a complete residual may make the physics constraint beneficial.

Bibliography

[1] Alix Benoit, T. Ivas, Mateusz Papierz, Asel Sagingalieva, Alexey Melnikov, and Elia Iseli. A fast and generalizable fourier neural operator-based surrogate for melt-pool prediction in laser processing. *arXiv preprint arXiv:2602.06241*, 02 2026. doi: 10.48550/arXiv.2602.06241.

[2] Rafael Bischof and Michael A. Kraus. Multi-objective loss balancing for physics-informed deep learning. *Computer Methods in Applied Mechanics and Engineering*, 439:117914, 2025. ISSN 0045-7825. doi: 10.1016/j.cma.2025.117914. URL https://www.sciencedirect.com/science/article/pii/S0045782525001860.

[3] T Chen and H Chen. Universal approximation to nonlinear operators by neural networks with arbitrary activation functions and its application to dynamical systems. *IEEE Trans. Neural Netw.*, 6(4):911–917, 1995.

[4] Woojin Cho, Minju Jo, Haksoo Lim, Kookjin Lee, Dongeun Lee, Sanghyun Hong, and Noseong Park. Parameterized physics-informed neural networks for parameterized PDEs. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp, editors, *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 8510–8533. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/cho24b.html.

[5] Salvatore Cuomo, Vincenzo Schiano Di Cola, Fabio Giampaolo, Gianluigi Rozza, Maziar Raissi, and Francesco Piccialli. Scientific machine learning through Physics–Informed neural networks: Where we are and what's next. *J. Sci. Comput.*, 92(3), September 2022.

[6] John Goldak, Aditya Chakravarti, and Malcolm Bibby. A new finite element model for welding heat sources. *Metall. Trans.*, 15(2):299–305, June 1984.

[7] Xiang Han, Zhuang Qian, Xinyue Gao, Huaping Li, Zhongqing Peng, and Yu Long. Three-dimensional modeling and analysis of directed energy deposition melt pools based on physical information neural networks. *Appl. Sci. (Basel)*, 15(17):9401, August 2025.

[8] AmirPouya Hemmasian, Francis Ogoke, Parand Akbari, Jonathan Malen, Jack Beuth, and Amir Barati Farimani. Surrogate modeling of melt pool temperature field using deep learning. *Additive Manufacturing Letters*, 5:100123,

2023. ISSN 2772-3690. doi: 10.1016/j.addlet.2023.100123. URL https://www.sciencedirect.com/science/article/pii/S277236902300004X.

[9] Jie Hou, Ying Li, and Shihui Ying. Enhancing PINNs for solving PDEs via adaptive collocation point movement and adaptive loss weighting. *Nonlinear Dyn.*, 111(16):15233–15261, August 2023.

[10] Feilong Jiang, Min Xia, and Yaowu Hu. Physics-informed machine learning for accurate prediction of temperature and melt pool dimension in metal additive manufacturing. *3D Printing and Additive Manufacturing*, 11(4):1679–1689, 2024. doi: 10.1089/3dp.2022.0363. URL https://journals.sagepub.com/doi/abs/10.1089/3dp.2022.0363.

[11] Wensheng Li, Chuncheng Wang, Hanting Guan, Jian Wang, Jie Yang, Chao Zhang, and Dacheng Tao. Generative adversarial physics-informed neural networks for solving forward and inverse problem with small labeled samples. *Computers & Mathematics with Applications*, 183:98–120, 2025. ISSN 0898-1221. doi: 10.1016/j.camwa.2025.01.025. URL https://www.sciencedirect.com/science/article/pii/S089812212500032X.

[12] Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Fourier neural operator for parametric partial differential equations. *arXiv preprint arXiv:2010.08895*, 2020.

[13] Zongyi Li, Hongkai Zheng, Nikola Kovachki, David Jin, Haoxuan Chen, Burigede Liu, Kamyar Azizzadenesheli, and Anima Anandkumar. Physics-informed neural operator for learning partial differential equations. *ACM / IMS J. Data Sci.*, 1 (3):1–27, July 2024.

[14] Lu Lu, Pengzhan Jin, Guofei Pang, Zhongqiang Zhang, and George Em Karniadakis. Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators. *Nat. Mach. Intell.*, 3(3):218–229, March 2021.

[15] Manav Manav, Nathanaël Perraudin, Yunong Lin, Mohamadreza Afrasiabi, Fernando Perez-Cruz, Markus Bambach, and Laura De Lorenzis. MeltpoolINR: predicting temperature field, melt pool geometry and their rate of change in laser powder bed fusion. *J. Intell. Manuf.*, October 2025.

[16] Xuhui Meng and George Em Karniadakis. A composite neural network that learns from multi-fidelity data: Application to function approximation and inverse pde problems. *Journal of Computational Physics*, 401:109020, 2020. ISSN 0021-9991. doi: 10.1016/j.jcp.2019.109020. URL https://www.sciencedirect.com/science/article/pii/S0021999119307260.

[17] Anadi Mondal and Subash L. Sharma. Bayesian neural network for critical heat flux prediction: a unified approach with uncertainty quantification. *Applied Thermal Engineering*, 278:127418, 2025. ISSN 1359-4311. doi: 10.1016/j.applthermaleng.2025.127418. URL https://www.sciencedirect.com/science/article/pii/S1359431125020101.

[18] Pavan Kumar Nalajam and Ramesh Varadarajan. A hybrid deep learning model for layer-wise melt pool temperature forecasting in wire-arc additive manufacturing process. *IEEE Access*, 9:100652–100664, 2021. doi: 10.1109/ACCESS.2021.3097177.

[19] Bohan Peng and Ajit Panesar. Multi-layer thermal simulation using physics-informed neural network. *Additive Manufacturing*, 95:104498, 2024. ISSN 2214-8604. doi: 10.1016/j.addma.2024.104498. URL https://www.sciencedirect.com/science/article/pii/S221486042400544X.

[20] M. Raissi, P. Perdikaris, and G.E. Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707, 2019. ISSN 0021-9991. doi: 10.1016/j.jcp.2018.10.045. URL https://www.sciencedirect.com/science/article/pii/S0021999118307125.

[21] Hesameddin Safari and Henning Wessels. Physics-informed surrogates for temperature prediction of multi-tracks in laser powder bed fusion. *arXiv preprint arXiv:2502.01820*, 2025.

[22] Hwijae Son, Sung Woong Cho, and Hyung Ju Hwang. Enhanced physics-informed neural networks with augmented lagrangian relaxation method (al-pinns). *Neurocomputing*, 548:126424, 2023. ISSN 0925-2312. doi: 10.1016/j.neucom.2023.126424. URL https://www.sciencedirect.com/science/article/pii/S0925231223005477.

[23] Zixue Xiang, Wei Peng, Xu Liu, and Wen Yao. Self-adaptive loss balanced physics-informed neural networks. *Neurocomputing*, 496:11–34, 2022. ISSN 0925-2312. doi: 10.1016/j.neucom.2022.05.015. URL https://www.sciencedirect.com/science/article/pii/S092523122200546X.

[24] Jibing Xie, Ze Chai, Luming Xu, Xukai Ren, Sheng Liu, and Xiaoqi Chen. 3D temperature field prediction in direct energy deposition of metals using physics informed neural network. *Int. J. Adv. Manuf. Technol.*, 119(5-6):3449–3468, March 2022.

[25] Qiming Zhu, Zeliang Liu, and Jinhui Yan. Machine learning for metal additive manufacturing: predicting temperature and melt pool fluid dynamics using physics-informed neural networks. *Comput. Mech.*, 67(2):619–635, February 2021.

Department of Mathematics
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden
www.chalmers.se

CHALMERS
UNIVERSITY OF TECHNOLOGY