POLITECNICO DI TORINO
Repository ISTITUZIONALE

Bridging Physics and Data in Metal Powder Bed Fusion with Scientific Machine Learning

Original
Bridging Physics and Data in Metal Powder Bed Fusion with Scientific Machine Learning / Depaoli, F., Ponzio, F., Macii, E., Di Cataldo, S.. - In: JOURNAL OF INTELLIGENT MANUFACTURING. - ISSN 0956-5515. - (2026). [10.1007/s10845-026-02820-8]

Availability:
This version is available at: 11583/3008171 since: 2026-03-09T16:04:46Z

Publisher:
Springer

Published
DOI:10.1007/s10845-026-02820-8

Terms of use:

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright

(Article begins on next page)

02 November 2026

Journal of Intelligent Manufacturing
https://doi.org/10.1007/s10845-026-02820-8

Bridging Physics and Data in Metal Powder Bed Fusion with Scientific Machine Learning

Fabio Depaoli[1] · **Francesco Ponzio**[1] · **Enrico Macii**[2] · **Santa Di Cataldo**[1]

Received: 17 November 2025 / Accepted: 8 February 2026
© The Author(s) 2026

Abstract

Metal Powder Bed Fusion (PBF) epitomizes the complexity of modern manufacturing, where strongly coupled thermal, mechanical, and fluid-dynamic interactions evolve over multiple spatial and temporal scales. Capturing these multiscale phenomena with predictive fidelity remains a central barrier to reliable, repeatable, and certifiable metal additive manufacturing. Conventional numerical solvers such as finite element or finite volume methods provide physical rigor but are computationally prohibitive for design exploration, uncertainty quantification, or real-time control. This paper positions *Scientific Machine Learning* (SciML) as a unifying paradigm that bridges the interpretability of physics-based modeling with the adaptability of data-driven inference. By embedding governing physical laws directly into learning procedure, or introducing biases into the architectures, SciML enables models that are both data-efficient and physically consistent, capable of accelerating high-fidelity simulations and supporting in-situ process optimization. The review is structured around three persistent challenges: mathematical complexity stemming from coupled multiphysics interactions, training instability inherent to physics-informed optimization, and practical integration barriers that limit industrial deployment. We analyze how emerging formulations such as Physics-Informed Neural Networks and Neural Operators address these challenges and advance toward robust, generalizable process surrogates. Beyond synthesizing existing methods, the paper outlines a roadmap toward *physics-grounded digital twins*, signaling a transformative step toward intelligent, self-correcting metal additive manufacturing systems.

Keywords Metal Additive Manufacturing · Powder Bed Fusion · Scientific Machine Learning · Physics-Informed Neural Networks · Neural Operators · Digital Twins

Introduction

Additive Manufacturing (AM) encompasses a family of processes that produce three-dimensional objects directly from digital models using a wide range of materials and technological principles. These processes can be classified according to criteria such as the initial material state, the layer formation mechanism, and the interlayer bonding strategy (Tony et al., 2023). Although these techniques fabricate components by superimposing successive material layers through different physical mechanisms, they all share a common digital workflow. In particular, the process starts from a Computer-Aided Design (CAD) model of the object, which is converted into a sequence of ultra-thin layers through an intermediate triangular mesh representation (Di Cataldo et al., 2021; Montalti et al., 2024).

In this context, the AM technologies that employ metallic materials, referred to as Metal AM (MAM), are increasingly adopted in industry as an alternative to traditional subtractive techniques, due to their ability to produce complex, lightweight, and highly customized metal components. More specifically, the selective deposition of metallic material only

✉ Fabio Depaoli
fabio.depaoli@polito.it

Francesco Ponzio
francesco.ponzio@polito.it

Enrico Macii
enrico.macii@polito.it

Santa Di Cataldo
santa.dicataldo@polito.it

[1] Department of Computer and Control Engineering, Polytechnic of Turin, Corso Duca Degli Abruzzi, 24, Torino 10129, Italy

[2] Interuniversity Department of Regional and Urban Studies and Planning, Polytechnic of Turin, Corso Duca Degli Abruzzi, 24, Torino 10129, Italy

Published online: 04 March 2026

where needed, which is at the core of the MAM approach, reduces metal waste and enables rapid design iteration without the need for tooling. This is especially attractive for aerospace (Liu et al., 2017), automotive (Goehrke, 2018), and biomedical applications (Javaid & Haleem, 2018; Aimar et al., 2019; Martorelli et al., 2020), which often involve complex and highly customized designs.

The two most widely adopted processes for MAM are Powder Bed Fusion (PBF) and Direct Energy Deposition (DED). While both enable layer-wise fabrication of metal components, they differ substantially in deposition strategy, resolution, and process dynamics:

- PBF spreads fine metal powder in thin layers and selectively fuses it using a laser or electron beam, resulting in high geometric resolution and excellent surface quality.
- DED deposits metallic material, either in powder or wire form, directly into the melt pool, enabling higher build rates and the fabrication of larger parts, albeit with reduced geometric precision.

Despite these differences, both technologies offer significant advantages over traditional manufacturing methods. In particular, they provide exceptional design freedom and allow the metallurgical properties of the final component to be tailored through careful adjustment of process parameters during fabrication (Lin et al., 2022). As a result, industrial interest in and adoption of MAM technologies have increased steadily in recent years.

However, the fabrication process in MAM is governed by complex and tightly coupled physical phenomena, making precise process control difficult. This complexity poses major challenges in terms of reliability and repeatability, which remain critical barriers to large-scale industrial deployment (Di Cataldo et al., 2021).

To address these challenges, research has increasingly focused on the development of *closed-loop control* strategies based on feedback loops (Liu et al., 2020). These approaches typically integrate: (i) measurements acquired from in-situ monitoring systems, and (ii) prior knowledge in the form of physics-based and data-driven models.

Within a closed-loop framework, *simulation* plays a central role as the predictive computational tool that links these information sources. By exploiting both real-time measurements and prior knowledge, simulations enable the prediction of process evolution and the anticipation of the system response to changes in process parameters, thereby supporting optimization and control decisions. In this context, simulation-based models naturally extend toward the Digital Twin(DT) concept: that is, a comprehensive virtual representation of a system that continuously evolves through the integration of real-time sensor data and operational feedback (Grieves & Vickers, 2017; Tao et al., 2019).

In manufacturing, and particularly in MAM, a DT integrates physics-based and data-driven models describing material behavior during fabrication with virtual representations of the manufacturing equipment and the design workflow (Kritzinger et al., 2018). This integrated virtual model enables monitoring and predictive capabilities, while supporting the testing and deployment of control strategies directly on the physical production system (Lu et al., 2020). To operate effectively, a DT must continuously acquire data from the real process, update its internal models accordingly, and generate appropriate responses to meet predefined objectives.

Enabling such closed-loop functionality requires simulation models that operate in (near) real time and are therefore computationally efficient, adaptive, and sufficiently accurate. When these conditions are satisfied, DTs can improve process consistency, enhance performance, and accelerate the transition toward reliable and intelligent metal additive manufacturing (Bayat et al., 2021), with demonstrated benefits also in sectors such as aerospace, energy, and manufacturing. Representative applications include predictive maintenance of aircraft engines (Tuegel et al., 2011), production-line optimization within Siemens' digital manufacturing platforms (Uhlemann, T.H.- J., Lehmann, C., Steinhilper, R., 2017), and process-parameter optimization in manufacturing systems (Annamalai & Nampoothiri, 2024b, a; Hu et al., 2025).

Traditionally, MAM simulations have been based on numerical techniques such as the Finite Element Method (FEM) and Finite Volume Method (FVM). These methods offer rigorous and physically consistent approximations of the governing Partial Differential Equation (PDEs), allowing detailed insights into the thermal, mechanical, and metallurgical behavior of the process. However, they also come with significant drawbacks: high computational demands, intricate meshing procedures, and the need for repeated calculations whenever the design or process parameters are modified. Such limitations make their integration into feedback loops or real-time optimization frameworks particularly challenging. Yet these are precisely the capabilities required for effective DT implementations.

Real-time and adaptive capabilities are key features of Scientific Machine Learning (SciML) simulation techniques. By combining first-principles models with machine learning (for example, neural networks), SciML provides a powerful and versatile framework, particularly well-suited to scenarios in which data availability is limited. In these approaches, prior physical knowledge is not imposed as an external constraint but is explicitly embedded within the learning process, the surrogate model architecture, or the training data (Cicirello, 2024). This integration creates a natural bridge between conventional numerical simulations and purely data-driven models, enabling the representation of

complex system behavior with high fidelity while achieving improved computational efficiency and real-time adaptability.

While the use of neural networks for solving differential problems has a long history (Lagaris et al., 1998; Psichogios & Ungar, 1992), their modern and most known formulation has emerged only recently in the form of Physics-Informed Neural Network (PINNs) (Raissi et al., 2019). These models approximate the solution of differential problems by sampling points within the computational domain and minimizing a loss function that quantifies the mismatch between the network predictions and the governing physical equations. In doing so, PINNs not only enhance computational efficiency but also provide a natural means of integrating physical laws with experimental observations. A complementary and increasingly influential research direction is represented by Neural Operator (NOs) (Anandkumar et al., 2019), which aim to learn parameterized solution operators for families of related problems using only a limited amount of simulation data.

Within the context of MAM, both PINNs and NOs have shown great promise for the efficient prediction of complex thermal and mechanical fields that arise during fabrication. This capability supports both process and geometry optimization while significantly reducing reliance on expensive, repeated numerical simulations. Ultimately, these advances are pivotal for the realization of DTs capable of interacting in real time with manufacturing systems. By combining physics-based knowledge with measured data, such models enable the creation of fast, adaptive simulation frameworks that can drive monitoring, optimization, and control feedback loops as schematically described in Figure 1 (Qi et al., 2021).

Motivation and contribution

Despite the significant advantages offered by SciML, their adoption in real-world applications remains limited and is hindered by several challenges:

1. *Mathematical complexity.* Real-world MAM processes involve multiphysics, multiscale, and time-dependent interactions, complicating both modeling and physics-informed learning (Jia et al., 2024).
2. *Training instability.* Physics-informed optimization typically results in highly non-convex, multi-objective problems, which can hinder convergence and stability (Anagnostopoulos et al., 2026).
3. *Implementation and usability barriers.* Most existing frameworks demand expertise in mathematical modeling, physics, and machine learning—skills rarely found together among process engineers or materials scientists.

Motivated by these observations, this paper presents a structured overview of state-of-the-art SciML techniques for MAM. By organizing our review around the three challenges identified above, we highlight current capabilities, existing gaps, and future opportunities for developing fast, physics-informed simulation models, with particular emphasis on the PBF process.

The PBF technique is inherently complex, involving tightly coupled multiphysics interactions (thermal, mechanical, fluid-dynamic, and optical), multiscale behaviors ranging from individual powder particles to full-part thermal cycles, and strongly time-dependent phenomena operating across disparate temporal scales. These characteristics make PBF exceptionally challenging to model with high accuracy. At the same time, they establish it as an ideal benchmark for testing and advancing physics-informed machine learning approaches. This duality provides a strong motivation for our review and underscores the importance of integrating physical insight with data-driven modeling to achieve predictive, efficient, and generalizable simulations.

Position within existing literature

Beyond general reviews of scientific machine learning such as Karniadakis et al. (2021), and broader overviews of physics-informed machine learning in manufacturing contexts, more specialized surveys have recently emerged: dedicated reviews of PINNs and NOs (Yang et al., 2024; Farea et al., 2024; Toscano et al., 2024; Liu et al., 2025; Patel et al., 2025) provide algorithmic taxonomies and cross-domain benchmarks. In the domain of metal additive manufacturing, prior surveys tend to focus on data-driven monitoring, process–structure–property modeling, microstructure prediction or material-specific applications rather than the simulation and control of the build process itself (Farrag et al., 2025; Hildebrand et al., 2024).

In contrast, the originality of this survey lies in its simulation-centric and process-level perspective. While existing surveys tend to focus on so-called *inverse problems* (i.e., predicting unknown model parameters from the integration of observed data and physical priors), our work concentrates on *forward problems*, where physics-informed approaches are employed to solve the differential equations governing the physical phenomena (Cicirello, 2024). In particular, we investigate PINNs and NOs, and on how these methods can be ultimately applied to accelerate the simulation of coupled thermo-fluid and mechanical behavior of PBF process, as preliminary proved in Zhu et al. (2021). Beyond summarizing the current algorithmic progress, we examine the mathematical and computational challenges inherent in multiphysics PBF modeling, analyze the training and stability bottlenecks of current formulations, and assess the software frameworks and technology-readiness required for industrial deployment.

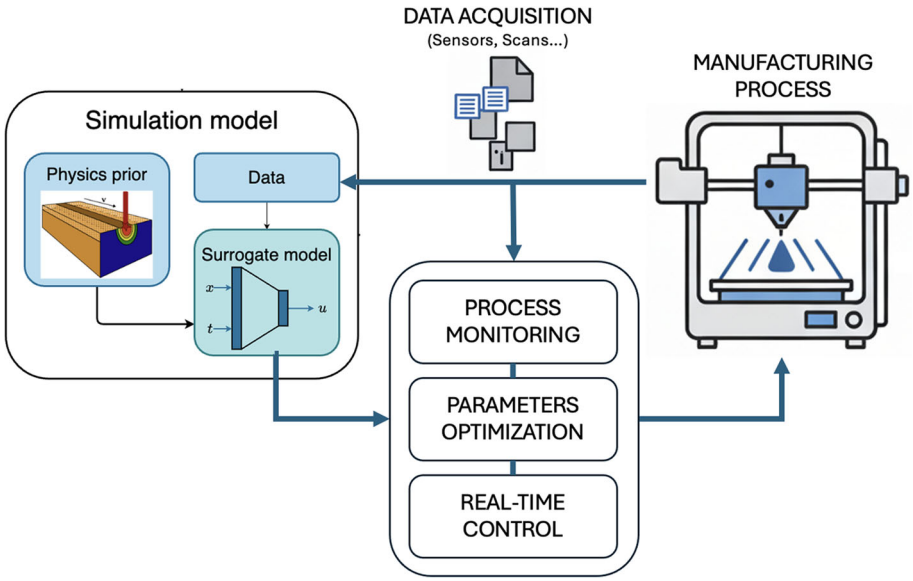


Fig. 1 Schematic of a Digital Twin with SciML-based simulation. Sensor data from the physical process are fed into a virtual surrogate model integrating physical priors. The surrogate enables rapid process simulations, supporting real-time adjustments to the machine and facilitating continuous process control and optimization

This *simulation-to-implementation* view differentiates our review from prior surveys, which typically emphasize either theoretical developments or application taxonomies. By connecting methodological advances in SciML with the practical requirements of autonomous and intelligent metal additive manufacturing, we identify SciML as a central enabling technology for *digital-twin-driven PBF*.

In line with this positioning, the present work is not a meta-analysis, but adopts a state-of-the-art review perspective aimed at synthesizing representative methodological contributions in SciML for metal PBF. The focus of the discussion is on modeling formulations, training challenges, and pathways toward integration into intelligent manufacturing systems, rather than on exhaustive literature enumeration or statistically driven evidence aggregation. Given the predominantly methodological and simulation-based nature of the reviewed studies, no formal quality or risk-of-bias scoring framework was applied; instead, methodological assumptions, limitations, and maturity levels are discussed qualitatively throughout the paper, with representative works highlighted for each thematic area.

This work does not propose a new digital twin definition or architecture; instead, it reviews and organizes recent advances in Scientific Machine Learning as enabling simulation technologies for physics-grounded, real-time digital twins in metal additive manufacturing.

Design and organization of the review

The literature discussed in this review was identified through searches conducted in Web of Science, Scopus, and Google Scholar. Web of Science and Scopus were selected for their comprehensive coverage of peer-reviewed engineering and manufacturing research, while Google Scholar was used to capture recent and emerging contributions in the rapidly evolving SciML domain. The review primarily considers works published between 2017 (year of advent of PINNs) and 2025, with earlier seminal studies included where necessary to establish methodological foundations.

Search queries combined key SciML terms with MAM keywords using Boolean operators. The core search query was (“physics-informed neural network*” OR “neural operator*”) AND (“powder bed fusion” OR “metal additive manufacturing”) to identify works on the application and we removed the constraints on the application for deeper research. Following database searches, duplicate records were removed and titles and abstracts were screened to exclude clearly out-of-scope studies and those that do not introduce novelties for what concern the algorithms while applied on topics different from MAM. Full-text articles were subsequently assessed for thematic relevance based on their methodological focus, applicability to metal additive manufacturing processes, and ability to address the computational challenges of the underlying mathematical models.

Overall, the review synthesizes approximately 70 peer-reviewed studies selected for their relevance to SciML-based forward modeling and simulation of metal PBF, closely related additive manufacturing processes, and models that can be applied on PBF simulations in principle because they have been developed for general physical phenomena. Studies focusing exclusively on inverse problems, purely data-driven black-box models without physical constraints, or non-metal additive manufacturing processes without transferable physics were excluded. Given the predominantly methodological and simulation-based nature of the reviewed studies, no formal quality or risk-of-bias scoring framework was applied; instead, methodological assumptions,

limitations, and maturity levels are discussed qualitatively throughout the paper.

The remainder of this paper is organized as follows. Section 2 outlines the background of SciML, presenting the formulations of PINNs and NOs. Section 3 addresses the mathematical modeling of PBF processes and the associated computational challenges that make accurate simulation difficult for both traditional numerical methods and SciML. Section 4 reviews the state of the art in SciML-based simulations for PBF models, while Section 5 discusses the available approaches to mitigate training challenges. Section 6 examines the current implementation of SciML algorithms in real-world applications, available software frameworks and integration pathways toward industrial deployments. Finally, Section 7 summarizes open problems and future research directions.

Background of Scientific Machine Learning

SciML refers to a broad class of approaches that integrate data-driven machine learning techniques with principles from computational science (Cicirello, 2024). By combining physical priors with information extracted from measured data, these methods aim to exploit the complementary strengths of both paradigms, leading to improved generalizability, flexibility, and computational efficiency. From a modeling perspective, SciML can be interpreted as a modern, algorithmic formalization of classical gray-box or semi-empirical modeling strategies, enabled by recent advances in machine learning and numerical computing.

As anticipated in the previous section, SciML algorithms are commonly categorized according to the type of problem they address *forward* or *inverse problems* (Cuomo et al., 2022). Forward problems focus on solving differential equations that model physical phenomena, whereas inverse problems aim to infer models or unknown parameters from data, incorporating different levels of physical prior knowledge, from purely data-driven (black-box) approaches to gray- and white-box formulations based on known physical models.

As our primary objective is the efficient simulation of the PBF process while retaining the ability to incorporate measurements during simulation, in the following sections we focus exclusively on forward problems. Within this context, PINN and NO represent the two families of SciML algorithms that best satisfy these requirements (Cuomo et al., 2022).

Physics-informed Neural Networks

PINNs, and physics-informed learning more broadly, provide a unified framework for combining physical laws and observational data within a single model (Raissi et al., 2019). In this approach, a neural network takes as input the spatio-temporal coordinates within the domain and outputs the physical quantities of interest, describing the state or evolution of the system. Training the neural network requires minimizing a loss function that aggregates multiple objectives into a single scalar quantity. These typically include: (i) a physics-based loss term that enforces the PDEs governing the underlying behavior; and (ii) terms ensuring that boundary and initial conditions are satisfied. Additionally, if data are available, the loss function includes a supervised loss term.

To mathematically define a PINN, consider a generic time-dependent problem on a spatial domain Ω over the time interval $[0, T]$. The governing equations are:

$$\begin{aligned} &\mathcal{N}[u](x,t) = f(x,t) \quad (x,t) \in \Omega \times (0,T] \\ &\mathcal{B}[u](x,t) = g(x,t) \quad (x,t) \in \partial\Omega \times (0,T] \\ &u(x,t) = h(x) \quad (x,t) \in \Omega \times \{0\} \end{aligned} \tag{1}$$

Here, $\mathcal{N}$ denotes a differential operator that may include time derivatives, and $\mathcal{B}$ is the boundary operator. The goal of a PINN is to approximate the solution u using a neural network.

In this formulation, $x \in \Omega \subset \mathbb{R}^d$ refers to the spatial coordinates (in one, two, or three dimensions), and $t \in [0, T]$ represents time. The function $u(x, t)$ corresponds to the physical field of interest (e.g., temperature, displacement, or stress), whose behavior is governed by $\mathcal{N}$. The functions f, g, and h are assumed to be known in forward problems, either everywhere in their domains or only at discrete data points.

Figure 2 provides a conceptual overview of the training process of a generic PINN.

PINNs approximate the solution $u : \bar{\Omega} \to \mathbb{R}$ of Equation (1) with a neural network u_θ parameterized by weights θ. The optimal parameters are obtained by minimizing the loss function

$$\mathcal{L}(\theta) = \lambda_r \mathcal{L}_r(\theta) + \lambda_b \mathcal{L}_b(\theta) + \lambda_0 \mathcal{L}_0(\theta) \tag{2}$$

where the individual components are defined as

$$\mathcal{L}_r(\theta) = \frac{1}{N_r} \sum_{i=1}^{N_r} \left| \mathcal{N}[u_\theta](x_r^{(i)}, t_r^{(i)}) - f(x_r^{(i)}, t_r^{(i)}) \right|^2 \tag{3}$$

$$\mathcal{L}_b(\theta) = \frac{1}{N_b} \sum_{i=1}^{N_b} \left| \mathcal{B}[u_\theta](x_b^{(i)}, t_b^{(i)}) - g(x_b^{(i)}, t_b^{(i)}) \right|^2 \tag{4}$$

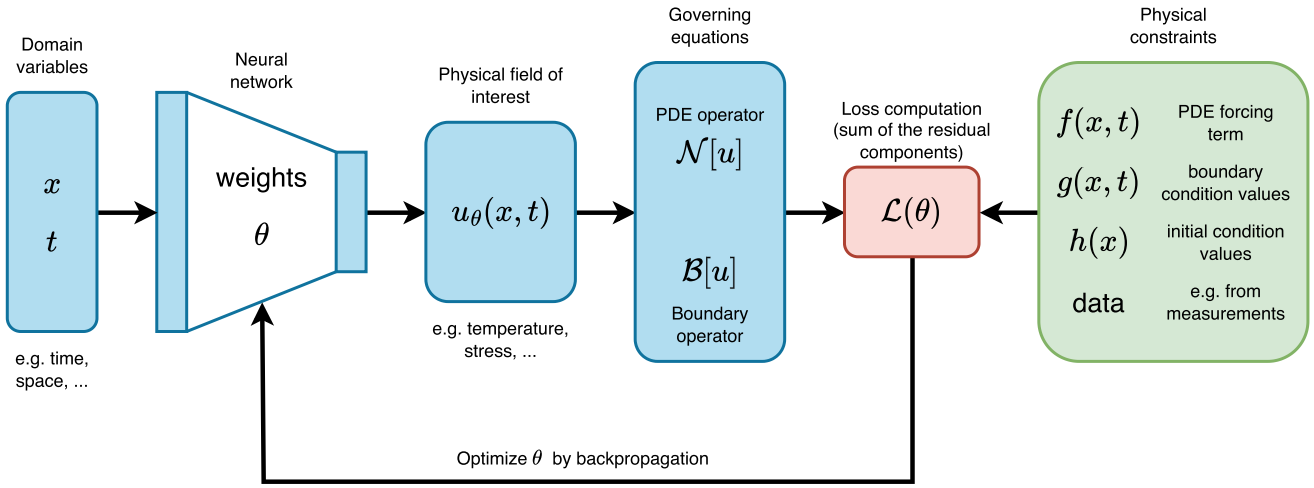


Fig. 2 Overview of the training process of a PINN. The neural network maps the spatio-temporal coordinates (x, t) to the predicted field $u_\theta(x, t)$. Automatic differentiation is used to evaluate the governing differential operators, enabling the explicit incorporation of physical constraints derived from the PDE, boundary conditions, initial conditions, and available measurements data. These constraints are enforced through the composite loss function $\mathcal{L}(\theta)$, defined by the residual terms in Equations (2)–(5), which is minimized via backpropagation to update the network weights θ

$$\mathcal{L}_0(\theta) = \frac{1}{N_0}\sum_{i=1}^{N_0}\left|u_\theta(x_0^{(i)}, 0) - h(x_0^{(i)})\right|^2 \tag{5}$$

When the functions f, g, and h are available throughout their entire domains, the sets of points $\left\{(x_r^{(i)}, t_r^{(i)})\right\}_{i=1}^{N_r}$, $\left\{(x_b^{(i)}, t_b^{(i)})\right\}_{i=1}^{N_b}$, and $\left\{x_0^{(i)}\right\}_{i=1}^{N_0}$ are randomly sampled within $\Omega \times (0, T]$, $\partial\Omega \times (0, T]$, and Ω, respectively. Otherwise, the collocation points are selected at the available measurement locations.

From a physical perspective, the individual components of the PINN loss function in Equations (3)-(5) enforce different aspects of the governing physics. The *residual loss* $\mathcal{L}_r(\theta)$ measures the degree to which the neural-network approximation satisfies the governing differential equations at collocation points within the domain. In the context of PBF thermal simulations, this term reflects the adherence of the model to the heat conduction equation, while in coupled thermo-fluid formulations it may additionally enforce momentum and mass conservation through the Navier–Stokes' equations.

The *boundary loss* $\mathcal{L}_b(\theta)$ penalizes violations of boundary conditions and is used to impose physically meaningful constraints such as heat convection and radiation at free surfaces, laser heat input, or flow conditions at melt-pool boundaries.

Finally, the *initial-condition loss* $\mathcal{L}_0(\theta)$ enforces consistency with the prescribed initial state of the system. For thermal problems, this term typically represents the initial temperature distribution of the powder bed or substrate before laser irradiation.

In case additional experimental measurements or simulated data are also available, it is possible to include a *supervised loss term* to further guide training.

Boundary and initial conditions may be enforced directly in the neural network structure. For example, the initial condition of Equation (1) can be imposed as a hard constraint by defining

$$u_\theta(x, t) = g(x) + a(t)\, N N_\theta(x, t), \tag{6}$$

where $a(t)$ is a prescribed function satisfying $a(0) = 0$. With this design, the initial condition $u(x, 0) = g(x)$ is always satisfied, because the added term vanishes at $t = 0$. This construction ensures exact enforcement of the constraint and eliminates the need for the corresponding loss term.

Training challenges of PINNs

Although the idea of developing a simulation model that is both fast in inference and capable of integrating data with physical laws is highly promising, the training of a PINN still presents several challenges that can prevent convergence to the correct solution. In particular, three major sources of training failure can be identified: (i) the imbalance among the different components of the loss function, (ii) the spectral bias inherent to neural networks, and (iii) the ill-posed nature of the residual loss in time-dependent problems.

Loss components imbalance

Training a PINN can be formulated as a multi-objective optimization problem, where the aim is to simultaneously

minimize the residual loss and the losses associated with boundary and initial conditions. In the original PINN formulation, this inherently multi-objective problem was simplified through *scalarization*: that is, by combining all objectives into a single loss function expressed as a weighted sum of the individual components. Although this approach achieved strong results on several classical benchmark problems, it often fails to generalize to more complex or stiff systems (Hao et al., 2024).

One of the first works to investigate the mechanisms behind these failures was presented in Wang et al. (2021a). The authors demonstrated that the gradient distributions corresponding to the different loss terms are often highly imbalanced, with the gradients of the boundary or initial losses typically being several orders of magnitude smaller than those of the residual loss. This imbalance usually arises because the residual term involves differential operators that amplify local errors, whereas boundary and initial condition terms act only on a limited subset of the domain. Additionally, differences in the number and distribution of collocation points for each loss component further exacerbate the problem.

Such imbalance leads to two primary consequences. First, the optimization process tends to prioritize minimizing the residual loss, thereby neglecting the accurate enforcement of boundary and initial conditions. As a result, the network struggles to propagate the influence of these constraints throughout the spatial and temporal domains, leading to non-physical or inconsistent solutions. Second, the disparity in gradient magnitudes induces *stiffness* in the optimization dynamics: a situation in which gradients evolve on widely different scales, forcing the optimizer to take very small steps to maintain stability. This stiffness can severely slow convergence, cause oscillatory training behavior, or, in extreme cases, drive the optimization trajectory toward divergence (Balduzzi et al., 2018).

Spectral bias

Spectral bias, also known as the *frequency principle*, refers to the tendency of neural networks to learn low-frequency (smooth) components of a target function much faster than high-frequency ones (Rahaman et al., 2019a). In practice, during training, a network first captures the broad, slowly changing structures of the solution, while finer details—such as sharp gradients, oscillations, or discontinuitiesare learned much more slowly or may not be represented accurately at all. In the context of PINNs, this bias poses a particular challenge because many physical systems exhibit multiscale behavior, where high-frequency components often carry essential information about complex dynamics (for example, wavefronts, shocks, or turbulent fluctuations). As a result, spectral bias can lead PINNs to converge toward overly smooth or physically inconsistent solutions.

Researchers often analyze the training dynamics of neural networks through the lens of the Neural Tangent Kernel (NTK) (Jacot et al., 2018). This theory establishes an equivalence between infinitely wide neural networks and Gaussian processes assuming an infinitesimal learning rate (i.e., $\frac{d\theta}{dv} = -\nabla\mathcal{L}(\theta)$ with v denoting the training time variable), providing a kernel-based framework that characterizes how the network outputs evolve during training. Considering the problem in Equation (1) and following the derivations in Wang et al. (2022), which formulate the NTK for a PINN, we obtain the following:

$$\begin{bmatrix} \frac{\partial u}{\partial v}[\mathbf{x}_0, \theta(v)] \\ \frac{\partial \mathcal{B}[u]}{\partial v}[\mathbf{x}_b, \theta(v)] \\ \frac{\partial \mathcal{N}[u]}{\partial v}[\mathbf{x}_r, \theta(v)] \end{bmatrix} = -K(v) \begin{bmatrix} u[\mathbf{x}_0, \theta(v)] - h(\mathbf{x}_0) \\ \mathcal{B}[u][\mathbf{x}_b, \theta(v)] - g(\mathbf{x}_b) \\ \mathcal{N}[u][\mathbf{x}_r, \theta(v)] - f(\mathbf{x}_r) \end{bmatrix}$$
$$K(v) = \begin{bmatrix} K_{00}(v) & K_{0b}(v) & K_{0r}(v) \\ K_{b0}(v) & K_{bb}(v) & K_{br}(v) \\ K_{r0}(v) & K_{rb}(v) & K_{rr}(v) \end{bmatrix} \tag{7}$$

Here, $K_{0b}(v) = K_{b0}^T(v)$, $K_{0r}(v) = K_{r0}^T(v)$, and $K_{br}(v) = K_{rb}^T(v)$. The vectors $\mathbf{x}_0$, $\mathbf{x}_b$, and $\mathbf{x}_r$ contain the spatial and temporal coordinates of all collocation points. The scalar products

$$(K_{00})_{ij}(v) = \left\langle \frac{du[x_0^{(i)}, \theta(v)]}{d\theta}, \frac{du[x_0^{(j)}, \theta(v)]}{d\theta} \right\rangle \tag{8}$$

$$(K_{0b})_{ij}(v) = \left\langle \frac{du[x_0^{(i)}, \theta(v)]}{d\theta}, \frac{d\mathcal{B}[u][x_b^{(j)}, \theta(v)]}{d\theta} \right\rangle \tag{9}$$

$$(K_{bb})_{ij}(v) = \left\langle \frac{d\mathcal{B}[u][x_b^{(i)}, \theta(v)]}{d\theta}, \frac{d\mathcal{B}[u][x_b^{(j)}, \theta(v)]}{d\theta} \right\rangle \tag{10}$$

$$(K_{0r})_{ij}(v) = \left\langle \frac{du[x_0^{(i)}, \theta(v)]}{d\theta}, \frac{d\mathcal{N}[u][x_r^{(j)}, \theta(v)]}{d\theta} \right\rangle \tag{11}$$

$$(K_{rr})_{ij}(v) = \left\langle \frac{d\mathcal{N}[u][x_r^{(i)}, \theta(v)]}{d\theta}, \frac{d\mathcal{N}[u][x_r^{(j)}, \theta(v)]}{d\theta} \right\rangle \tag{12}$$

$$(K_{br})_{ij}(v) = \left\langle \frac{d\mathcal{B}[u][x_b^{(i)}, \theta(v)]}{d\theta}, \frac{d\mathcal{N}[u][x_r^{(j)}, \theta(v)]}{d\theta} \right\rangle \tag{13}$$

define the components of the NTK $K(v)$, which describe how variations in the network parameters influence the outputs of u, $\mathcal{B}[u]$, and $\mathcal{N}[u]$ at the collocation points.

In Wang et al. (2022), the authors demonstrated that for a linear problem, u, $\mathcal{B}[u]$, and $\mathcal{N}[u]$ converge in probability to Gaussian processes as the number of neurons N in the

hidden layer tends to infinity. The resulting Gaussian process possesses a deterministic kernel K^*. When the activation function satisfies suitable continuity conditions, the kernel remains effectively constant during training, that is:

$$\lim_{N\to\infty} \sup_{v\in[0,V]} \|K(v) - K(0)\|_2 = 0. \tag{14}$$

Under these assumptions, we can approximate the training evolution as

$$\begin{bmatrix} u[\mathbf{x}_0, \theta(v)] \\ \mathcal{B}[u][\mathbf{x}_b, \theta(v)] \\ \mathcal{N}[u][\mathbf{x}_r, \theta(v)] \end{bmatrix} - \begin{bmatrix} h(\mathbf{x}_0) \\ g(\mathbf{x}_b) \\ f(\mathbf{x}_r) \end{bmatrix} \approx -e^{-K(0)v} \cdot \begin{bmatrix} h(\mathbf{x}_0) \\ g(\mathbf{x}_b) \\ f(\mathbf{x}_r) \end{bmatrix}$$
$$\approx -Qe^{-\Lambda v}Q \cdot \begin{bmatrix} h(\mathbf{x}_0) \\ g(\mathbf{x}_b) \\ f(\mathbf{x}_r) \end{bmatrix}. \tag{15}$$

The eigenvalues in Λ of $K(0)$ determine the convergence rate of each corresponding eigenvector in Q, which in turn governs the spectral content of the network output during training. Empirical studies have shown that larger eigenvalues typically correspond to lower-frequency components (Rahaman et al., 2019b; Basri et al., 2020; Ronen et al., 2019), thereby explaining why PINNs are affected by spectral bias. By modifying the NTK spectrum through spectral features enrichment (see Section 4) or through adaptive weighting of the loss components (see Section 5), the training convergence and robustness can be significantly improved.

Fixed-point attraction

In a general dynamical system governed by

$$u_t = f(u),$$

a *fixed point* u^* is a stationary state satisfying $f(u^*) = 0$; that is, once the system reaches u^*, it remains there indefinitely because $du/dt = 0$. Fixed points can be further classified based on their response to small perturbations: a fixed point is *stable* if nearby trajectories return to it over time, and *unstable* if such perturbations grow and the trajectories move away.

In Rohrhofer et al. (2023), the authors show that neural network parameterizations yielding solutions that coincide with the fixed points of the associated dynamical system also correspond to minima of the residual (physics-informed) loss. The reasoning behind the result in Rohrhofer et al. (2023) can be summarized as follows. If the neural network output satisfies $u_\theta = u^*$, then $f(u_\theta) = 0$, meaning that the residual of the governing equation vanishes. For neural networks with a finite number of layers and neurons and with continuous activation functions, the mapping $\theta \mapsto u_\theta$ is continuous. Hence, perturbations of the volume θ^*, i.e. the parameters vector such that $u_{\theta^*} = u^*$, results in a negligible change in the physics loss. Therefore, θ^* acts as a *global minimizer* of the residual loss and is surrounded by a non-trivial *basin of attraction*: that is, the region D in the parameter space in which the training algorithm converges to θ^* for each initial configuration $\theta_0 \in D$.

This property implies that both stable and unstable fixed points of the physical system can attract the optimization dynamics of the PINN. In practice, the training process may converge toward parameterizations that reproduce such fixed points, even if the corresponding solutions are physically meaningless. In this case, the network becomes trapped in a non-physical equilibrium that satisfies the loss minimization criteria but fails to represent the correct system dynamics.

Furthermore, the study in Rohrhofer et al. (2023) showed that the likelihood and severity of fixed-point attraction depend mainly on two factors: the initial condition u_0 and the simulation horizon T. The influence of the initial condition is straightforward: if u_0 lies close to a fixed point, its contribution to the total loss, through both the initial condition term and the local residuals, becomes negligible. This reasoning also clarifies the dependence on the simulation time window: as T increases, the relative influence of the initial condition term within the total loss decreases. Consequently, parameterizations corresponding to spurious fixed-point solutions tend to dominate the optimization landscape for larger T, making the network more prone to converge toward non-physical global minima.

Neural Operators

PINNs learn solutions of differential equations by enforcing the governing physics directly through the loss function. NOs, on the other hand, aim to learn mappings between function spaces. Therefore, they can be used to predict solutions of related problems without explicitly solving the governing equations each time.

The objective of NOs is to learn mappings between infinite-dimensional function spaces using input-output data pairs (Anandkumar et al., 2019). In other words, neural operators extend supervised learning from finite-dimensional vectors to operators acting on functions. More precisely, the aim is to approximate a mapping between two Banach spaces of functions, X and Y (that is, a complete metric space of functions) using a parametric model

$$\mathcal{G} : X \times \Theta \to Y \tag{16}$$

where Θ is a finite-dimensional parameter space. In practical computations, the spaces X and Y are discretized so that their elements can be represented numerically. Neural operators can further be made *physics-informed* by defining

a loss function that incorporates the governing physics of the operator mapping from X to Y.

In their foundational work, Anandkumar et al. (2019) introduced the Graph Neural Operator (GNO), a NO designed to map an input function $a : \Omega \to \mathbb{R}^{d_a}$, with $\Omega \subseteq \mathbb{R}^d$, to the solution $u : \Omega \to \mathbb{R}^{d_u}$ of the differential problem

$$\begin{aligned} \mathcal{L}_a[u](x) &= f(x) \quad x \in \Omega \\ u(x) &= 0 \qquad x \in \partial\Omega \end{aligned} \tag{17}$$

which admits a solution of the form

$$U(x) = \int_\Omega G_a(x, y) f(y)\, dy \tag{18}$$

From a physical standpoint, the integral representation in Equation (18) provides an intuitive interpretation of the solution operator. The so-called *Green's function* $G_a(x, y)$ represents the response of the physical system governed by the operator $\mathcal{L}_a$ at the observation point x due to an instantaneous point source applied at location y. In the case of pure thermal simulations relevant to PBF processes, $G_a(x, y)$ corresponds to the temperature measured at position x resulting from a localized heat input at position y. The integral therefore represents the cumulative contribution of all distributed sources described by the function $f(y)$ over the domain Ω, consistent with the superposition principle in linear systems and analogous to the impulse response concept in linear time-invariant dynamical systems.

G_a is unique and continuous for $x \neq y$ under suitable assumptions (Evans, 2022), which motivates the use of neural networks for approximation. To capture the behavior of the integral operator, the authors proposed a multi-layer architecture in which each layer uses a message-passing graph neural network to approximate the integral:

$$v_{t+1}(x) = W v_t(x) + \frac{1}{|N(x)|} \sum_{y \in N(x)} k_\phi[x, y, a(x), a(y)] v_t(y) \tag{19}$$

where $N(x)$ denotes the neighborhood of x and k_ϕ is a neural network kernel acting on pairs of points.

The Fourier Neural Operator (FNO) (Li et al., 2021), in contrast, does not rely solely on local graph neighborhoods to approximate the integral operator. Instead, it exploits the Fourier transform, implemented efficiently through the Fast Fourier Transform (FFT), to capture both local and long-range interactions:

$$\int_\Omega k[x, y, a(x), a(y)] v_t(y)\, dy \approx \mathcal{F}^{-1}[R_\phi \cdot \mathcal{F}(v_t)](x) \tag{20}$$

where R_ϕ is a learnable operator applied in the Fourier domain.

DeepONet (Lu et al., 2021a) takes a different approach: it directly builds on the universal approximation theorem for operators (Chen & Chen, 1995). This theorem ensures that an operator can be learned using two neural networks. The first, called the *branch network*, encodes the discretized input function, while the second, the *trunk network*, encodes the evaluation location x. The operator output is then obtained as

$$\mathcal{G}[a](x) = \sum_{k=1}^{p} b_k(a)\, t_k(x) \tag{21}$$

where p is a tunable hyperparameter that controls the expressive power of the model.

Training challenges of NOs

Although NOs provide a complementary framework for learning mappings between function spaces, their training also presents a range of significant difficulties. Three prominent issues are: (i) the heavy requirements in terms of training data, (ii) the errors due to discretization, and (iii) the complex optimization dynamics, especially when physical constraints are included. These challenges share some similarities with those observed in PINNs, but their origins differ. Table 1 summarizes the key distinctions between the failure modes of PINNs and NOs.

High data requirements Unlike PINNs, which typically train a model for a specific instance of a differential problem, NOs aim to learn an entire mapping from input functions to output functions. This generality demands a broad and diverse set of input–output function pairs to cover the operator's variability. Acquiring such datasets often involves running many expensive numerical simulations, which limits scalability and practicality in data-scarce domains. For example, strategies for pretraining neural operators emphasize that performance is highly dependent on the choice and volume of training data as well as model–dataset alignment (Zhou et al., 2024).

Discretization mismatch and mesh-dependence Because NOs are formally defined in infinite-dimensional function spaces but trained on discretized approximations, their performance may degrade when the discretization (mesh, grid resolution, sampling pattern) changes between training and inference. Recent work explicitly studies the "discretization mismatch error" of neural operators–showing that training at coarse resolution and inferring at fine resolution does not guarantee preservation of accuracy (Gao et al., 2025). Similarly, architectures that lack discretization-invariance can

suffer when applied to irregular meshes or multiscale geometries (Hao et al., 2023).

Optimization complexity and physics enforcement The optimization landscape of operator learning models is highly nonconvex and can be particularly challenging when incorporating physical constraints (e.g., boundary conditions, conservation laws, etc.). Architectures such as FNOs, which combine spectral filtering and nonlinear transformations, require careful mode selection to avoid under- or over-fitting of frequency components (George et al., 2024) . Furthermore, when physics-informed terms are added to the training objective, balancing the competing loss components becomes difficult, and the gradients can become stiff or unstable, limiting convergence and generalization (Cheng et al., 2025).

SciML versus classical numerical methods

While SciML approaches are relatively recent additions to the simulation landscape, classical numerical methods remain the backbone of computational physics and engineering. These methods refer to a family of physics-based techniques for solving partial differential equations (PDEs) through the explicit discretization of the governing equations and, when required, of the physical domain. Depending on the discretization strategy, they are commonly categorized into (i) *finite element methods* (FEM), which approximate the solution using local basis functions defined on a mesh; (ii) *finite volume methods* (FVM), which enforce conservation laws over discrete control volumes; and (iii) *spectral methods*, which represent the solution through global expansions in orthogonal basis functions. These approaches are fully established in every sector of engineering, and provide strong theoretical guarantees in terms of consistency, stability, and convergence (Hughes, 2003; Leveque, 2002; Boyd, 2001). This makes them particularly suitable for predictive simulations and design verification. On the other hand, the computational cost of these methods increases rapidly with problem dimensionality, geometric complexity, and multiphysics coupling, which limits their applicability in real-time or control-oriented settings (Belytschko et al., 2014; Quarteroni et al., 2014).

SciML approaches, including PINNs and NOs, address these limitations by embedding physical laws directly into machine-learning models. PINNs, as described earlier in this Section, incorporate the governing equations, boundary conditions, and initial conditions into the loss function of a neural network, enabling mesh-free approximations of PDE solutions (Raissi et al., 2019; Karniadakis et al., 2021). NOs, by contrast, learn mappings between infinite-dimensional function spaces and thus provide resolution-invariant surrogate models that generalize across different discretizations (Li et al., 2021; Kovachki et al., 2023). Once trained, these approaches enable fast inference and naturally support data assimilation and model adaptation, making them particularly attractive for DT implementations and closed-loop control applications (Rackauckas et al., 2021; Tao et al., 2019). Table 2 summarizes the key differences between the two methodological families.

Beyond their conceptual and algorithmic differences, traditional numerical methods and SciML approaches also differ substantially in their computational workflows (see Figure 3). Classical numerical methods follow a sequential pipeline that begins with geometry definition, proceeds through mesh generation and numerical solver configuration, and culminates in the computation of the solution. When the problem setup or parameter configuration changes, this workflow must typically be repeated in its entirety.

By contrast, PINNs eliminate the need for explicit mesh construction by sampling collocation points within the computational domain and training a neural network through a physics-informed loss function. When the governing parameters are spatially constant, as in isotropic problems, PINNs naturally support the construction of parametric models (Depaoli et al., 2024).

NOs adopt a fundamentally different strategy by learning a direct mapping from the parameter space (potentially including functional inputs) to the solution space. This approach relies on an offline training phase in which traditional numerical solvers generate high-fidelity solutions for a selected set of parameter configurations. These solutions form the training dataset for the NO, which can then rapidly predict solutions for previously unseen parameter settings.

Mathematical modeling of the PBF process

To develop accurate and efficient surrogate models for PBF processes, a thorough understanding of the underlying physics and of conventional simulation approaches is essential.

Modeling and simulation of PBF remain highly challenging due to the strong coupling between multiple physical phenomena acting over vastly different spatial and temporal scales (Jia et al., 2024). In PBF, a thin and uniformly spread layer of loose metallic powder forms a *powder bed* on top of the previously consolidated material, referred to as the *solid substrate*. A high-energy heat source, most often a laser (or in some cases an electron beam), then selectively fuses regions of this powder layer according to the part geometry. Local melting creates a small *melt pool* of liquid metal that follows the scan path and subsequently solidifies, bonding metallurgically to the underlying substrate. This layer-by-layer sequence progressively forms the component and enables the fabrication of highly complex three-dimensional geometries with fine resolution (Di Cataldo et al., 2021).

Table 1 Comparison of PINNs and NOs, with special focus on the main training challenges.

Aspect	PINNs	NOs
Objective	Learn a single solution of a PDE for given initial/boundary data	Learn the entire operator mapping input functions to solution functions
Main source of difficulty	Imbalance between residual and boundary/initial losses	Large and diverse datasets required to represent operator variability
Representation bias	Spectral bias: difficulty in capturing high-frequency dynamics	Sensitive to discretization choices; may not generalize across meshes
Optimization issues	Residual loss may be ill-posed and lead to convergence to spurious fixed points	Highly nonconvex optimization; physics enforcement can destabilize gradients
Generalization	Typically limited to similar problem settings and only inside the selected computational domain	Performance depends strongly on training data coverage of operator space

Table 2 Comparison between classical numerical methods and SciML approaches

Aspect	Classical numerical methods	PINNs/NOs
Mathematical foundation	Direct discretization of PDEs with strong theoretical guarantees	Learning-bases approximation constrained by physical laws
Spatial discretization	Mesh- or basis-dependent	Mesh-free (PINNs) or discretization invariant (NOs)
Computational cost	High for large-scale or multi-physics problems	High during training, low during inference
Real-time capabilities	Limited by solver complexity	Naturally suited for real-time inference and control
Generalization	Problem- and mesh-specific	Cross-geometry or cross-resolution generalization (NOs)
Data integration	Typically limited to boundary/initial conditions	Naturally integrates observational and sensor data
Accuracy and robustness	High accuracy with high resolution	Sensitive to training dynamics and data quality
Suitability for DTs	Computationally demanding for closed-loop use	Well aligned with adaptive, feedback-driven DTs

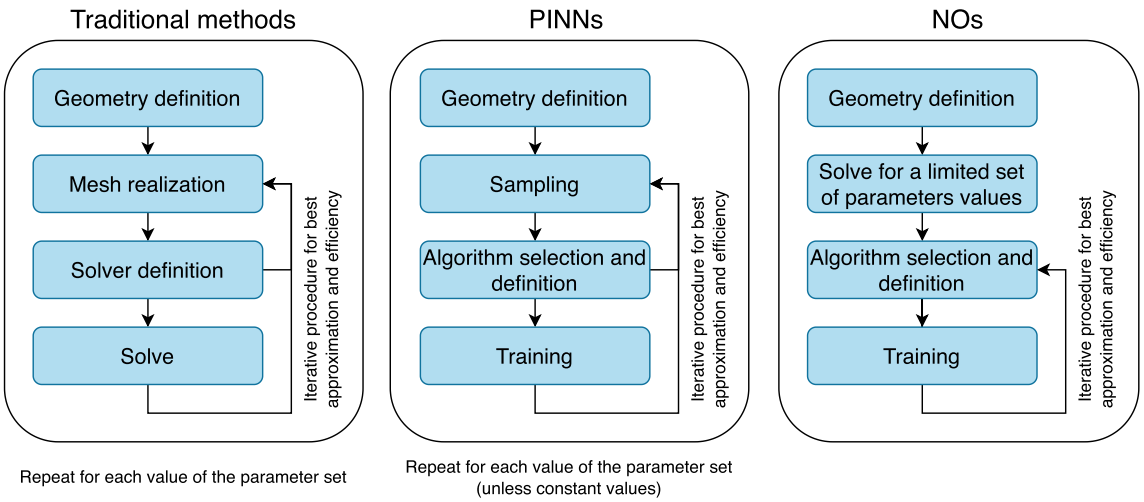


Fig. 3 Solution of differential problems: workflows of traditional numerical methods against PINNs and NOs

This process involves a large set of nonlinear, temperature-dependent, and tightly coupled phenomena, that are schematically in illustrated in Figure 4. While the discussion below primarily refers to laser-based PBF, many of the same physical mechanisms also apply to electron beam PBF, albeit with differences driven by the vacuum environment, higher preheat temperatures, and variations in melt-pool dynamics. These mechanisms include:

1. **Power source-powder interaction:** Absorption, multiple scattering, and reflection of the incident beam occur as energy penetrates a granular and partially transparent medium (i.e., the powder bed) with heterogeneous optical properties. These interactions depend on particle size, morphology, oxidation, and packing state, and govern the local distribution of heat.
2. **Powder packing characteristics:** Powder layers exhibit heterogeneous porosity and anisotropic conduction paths. Variations in local packing (spatial arrangement, packing density, contact points) can alter melt initiation thresholds and contribute to spatially varying thermal conductivity, energy absorption and defect susceptibility.
3. **Heat transfer within the powder bed:** Combined conduction through particle contact networks, radiation across voids, and convective transfer when melting begins. Thermal properties evolve dynamically as particles fuse and pores collapse.
4. **Energy coupling to solidified layers:** Heat diffusion into underlying material governs re-melting, heat-affected zones, and cross-layer microstructural evolution. Heat accumulation contributes to distortion at the part scale.
5. **Melt pool fluid dynamics:** Driven by strong temperature-dependent surface tension gradients (Marangoni convection), density variations that generate buoyancy forces, and recoil pressure resulting from metal evaporation, the liquid metal circulates vigorously within the melt pool. The balance between these forces governs the melt-pool dimensions, its ability to properly wet and fuse the underlying substrate, and its stability against process defects such as key-holing, balling of the liquid track, or particle ejection (spatter).
6. **Heat transfer in solidified material:** Cooling is dominated by conduction. Extremely rapid thermal gradients (up to 10^6 K/s) drive microstructural transformations, influencing grain morphology and final mechanical properties.
7. **Metal vaporization and plume dynamics:** At high energy densities, evaporation produces a recoil force that depresses the melt surface and can trap gas, causing porosity. Spatter and plume behavior differ between laser and electron beams due to gas flow versus vacuum conditions.
8. **Residual stress evolution:** Repeated heating-cooling thermal cycling induces non-uniform expansion and contraction, causing stress accumulation that may lead to warping, delamination, or cracking.

Directly capturing all these coupled effects simultaneously at the highest possible resolution would be impractical and prohibitively expensive. For this reason, modeling approaches are typically classified across three main scales: *part scale*, *mesoscale*, and *powder scale*, each adopting specific simplifications to balance computational cost and physical accuracy.

At the **part scale**, models focus on global heat transfer and residual stress evolution and their impact on metallurgic transformations. They typically employ thermo-mechanical equations while simplifying aspects related to powder behavior, energy delivery, and melt pool fluid dynamics. These simplifications often require extensive experimental calibration to provide reliable predictions.

At the **mesoscale**, melt-pool geometry and thermal gradients are resolved using continuum fluid mechanics. These models treat the material either as a continuum or a mass flux, enabling accurate representation of the temperature field. They also support local prediction of defect formation and melt-track quality.

At the **powder scale**, individual particles are explicitly represented to simulate the transient free-surface evolution responsible for keyhole porosity, splashing, and pore entrapment. Such high-fidelity models provide invaluable physical insights of the melt pool dynamics, including mass transfer, free-surface evolution and defect formation, but at enormous computational cost for the simulations.

Part-scale models

Part-scale models aim to predict the global temperature evolution during the build and its effects on microstructure and residual stress. They treat both the previously consolidated layers and the newly deposited powder as a homogenized continuum, which enables simulations at the part level but prevents accurate resolution of melt-pool fluid dynamics. Instead, melt behavior is captured only through boundary conditions or effective heat input models.

The thermal response is governed by the transient heat conduction equation:

$$\rho c \frac{\partial T}{\partial t} = \nabla \cdot (k \nabla T) + q, \tag{22}$$

where ρ, c, k, and T denote density, heat capacity, thermal conductivity, and temperature, respectively, and q represents the volumetric heat input from the beam.

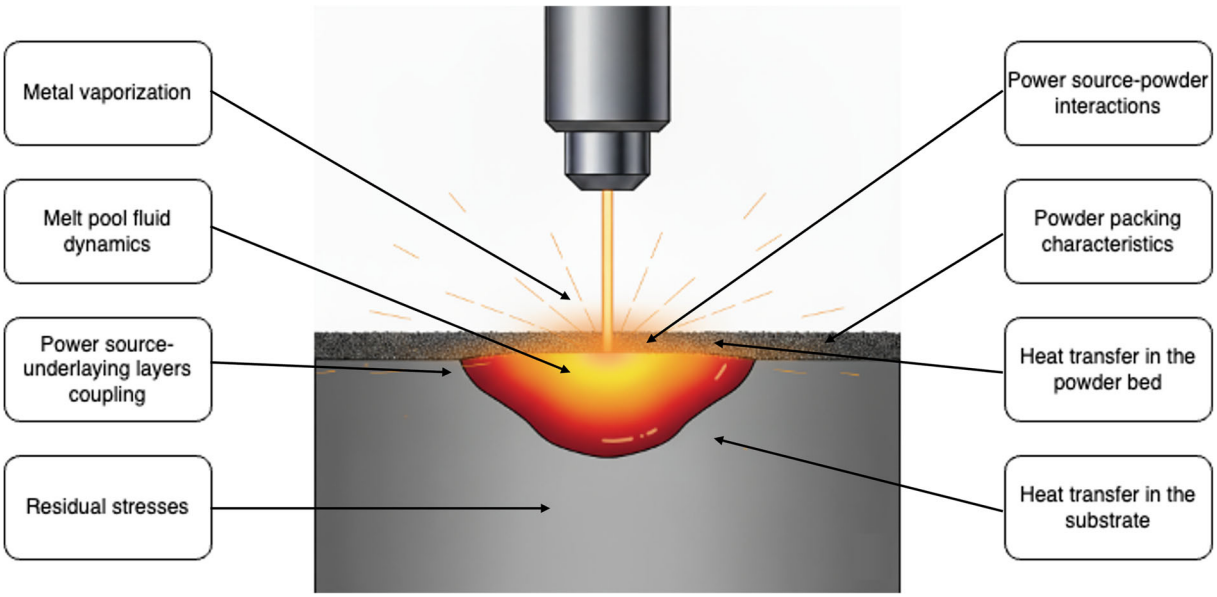


Fig. 4 Overview of the key physical interactions involved in a PBF process

Phase transformation effects are often incorporated using the apparent heat capacity method (Bonacina et al., 1973):

$$c_{\text{apparent}} = \begin{cases} c & T < T_s \\ c + \frac{L}{T_l - T_s} & T_s \leq T \leq T_l \\ c & T > T_l \end{cases} \tag{23}$$

with (T_s, T_l, L) the solidus temperature, liquidus temperature, and latent heat of fusion. This formulation provides a simplified representation of melting and solidification without explicitly tracking the melt pool.

Accurate prediction of q is crucial, as the shape, depth, and intensity of the heat source strongly affect thermal gradients. Analytical beam models (e.g., Gaussian, double-ellipsoidal) are therefore calibrated against experimental measurements to reproduce melt-pool dimensions (Rahman Chukkan et al., 2015). Similarly, continuum modeling of the powder requires estimating effective thermal and mechanical properties from bulk material characteristics and porosity (Denlinger et al., 2016).

Mechanical behavior is typically computed in a quasi-static framework by solving the equilibrium equation:

$$\nabla \cdot \sigma + f = 0, \tag{24}$$

where σ is the Cauchy stress tensor and f the body and surface forces. Residual stresses arise through the nonlinear coupling between elastic, plastic, and thermally induced strains, driven by steep gradients and repeated heating and cooling cycles.

Overall, part-scale models provide computationally tractable insight into thermal history and macroscopic distortion, but they rely on simplifications that limit their ability to capture local melt-pool physics.

Meso/powder-scale models

Meso- and powder-scale models aim to capture the detailed melt-pool behavior and the physics of laser–powder interaction that are simplified or neglected at the part scale. These models bridge the gap between the local phenomena governing defect formation and the larger-scale thermal response of the part.

The first step consists in simulating the interaction between the heat source and the powder bed. Two main strategies are typically used: (i) treating the powder as a continuum and modeling laser absorption similarly to part-scale heat source representations (Mukherjee et al., 2018), or (ii) explicitly representing individual particles and tracking beam–particle interactions using ray tracing or Monte Carlo methods (Willy et al., 2018). The latter approach improves accuracy in predicting the onset of melting and the shape of the melt pool, at a significantly higher computational cost.

Once melting initiates, the governing physics are described by the coupled thermo-fluid equations for mass, momentum, and energy conservation:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{v}) = 0, \tag{25}$$

$$\frac{\partial (\rho \mathbf{v})}{\partial t} + \nabla \cdot (\rho \mathbf{v} \otimes \mathbf{v}) = -\nabla p + \nabla \cdot \tau + S_{\text{mom}}, \tag{26}$$

$$\frac{\partial(\rho h)}{\partial t} + \nabla \cdot (\rho h \mathbf{v}) = \nabla \cdot (k \nabla T) + S_{\text{energy}}, \tag{27}$$

where ρ is the density, $\mathbf{v}$ the velocity field, p the pressure, h the specific enthalpy, and τ the viscous stress tensor. The source terms S_{mom} and S_{energy} account for recoil pressure due to evaporation and latent heat effects during melting and solidification.

Continuum formulations are commonly solved via the Navier–Stokes equations on moving domains, combined with free-surface tracking techniques. In contrast, powder-scale models treat the material as a collection of discrete particles (e.g., discrete element methods, lattice Boltzmann or particle-based multiphase solvers), enabling detailed study of wetting dynamics, particle entrainment, spatter formation, and pore entrapment.

While these models achieve the highest level of physical realism, they are typically limited to very small spatial domains and short timescales due to their extreme computational expense.

The tradeoff evident in Table 3 underscores the fundamental bottleneck: high-fidelity models are too expensive for design exploration or process optimization, while computationally efficient models often lack the resolution needed to predict defect formation.

Modeling challenges and impact on neural surrogates

Summarizing, the complex nature of PBF modeling poses significant challenges for the development of machine-learning-based surrogate models. These challenges can be primarily grouped into three categories:

1. **Multiphysics coupling.** Even when restricting attention to thermal phenomena, several physically distinct contributions must be incorporated into the loss function (e.g., conduction, boundary conditions, phase change), and achieving an effective balance between these terms during training is far from straightforward. When additional physics such as melt-pool fluid flow, vaporization, or stress formation are included, the coupling between losses becomes even more intricate and requires careful enforcement of physical consistency.
2. **Multiscale behavior.** Solution fields in PBF exhibit highly localized and rapidly evolving features. Sharp thermal gradients form within the narrow melt region, while the surrounding material undergoes smoother temperature variations at much larger scales. Moreover, the powder-scale physics introduces discrete–continuous interactions, evolving free surfaces, and topological changes such as pore entrapment or spatter, which violate standard smooth-function assumptions. This broad spectral and structural variability amplifies spectral bias and challenges neural networks to simultaneously capture high-frequency details and global trends.
3. **Strong time dependence.** The motion of the heat source induces microsecond-scale melting and solidification, whereas heat accumulation and residual stress develop over much longer time horizons. This separation of temporal scales creates stiffness in the governing equations and can cause physics-informed neural networks to collapse toward trivial or fixed-point solutions unless causality-aware or sequential training strategies are introduced.

Because models defined at different scales involve distinct physical phenomena, they also exhibit different subsets of the associated computational challenges. Some algorithms are tailored to specific data structures, such as vectors, matrices, or meshes, which may be more suitable for addressing particular aspects of the problem. Nevertheless, the overall workflow for employing these algorithms is largely independent of the specific application, as the training procedure does not differ from that used in purely data-driven approaches (see Section 2). The algorithms presented in the following sections address these challenges by targeting specific aspects of this variability.

Solving the modeling challenges of PBF with SciML methods

As discussed in Section 3, high-fidelity PBF simulations must capture several tightly coupled physical mechanisms with widely separated spatial and temporal scales. Traditional numerical solvers are capable of resolving these effects, but their high computational cost limits their applicability in design-space exploration, uncertainty quantification, or in-situ process control. Scientific Machine Learning (SciML) offers a promising alternative by leveraging both observational data and physical constraints to accelerate simulations while preserving predictive fidelity.

Although the primary focus of this survey is on PBF, dedicated SciML studies on this specific process remain scarce. For completeness, and due to the strong similarity of the governing equations, we therefore include in our review contributions demonstrated in other metal additive manufacturing techniques (e.g., DED) as well as more general multiphysics systems. The insights gained from these settings are largely transferable to PBF, since the same modeling hurdles arise: strong multi-physics coupling, pronounced multiscale behavior, and stiff time-dependent dynamics. In the following, we review how SciML methods have been used to address each of these challenges.

Table 3 Comparison of modeling scales for PBF.

Scale	Physics captured	Typical outputs	Computational cost
Part scale	Global heat transfer, approximate solidification, macroscopic stress formation, homogenized powder behavior	Temperature history, residual stresses, global distortion	Low to moderate
Mesoscale	Laser-powder interaction, melt-pool geometry, thermal gradients, defect formation mechanisms	Melt-pool dimensions, lack-of-fusion prediction, bead continuity	High
Powder scale	Particle-level melting, spatter, pore formation, free-surface evolution, keyhole dynamics	Detailed melt-pool fluid flow and topology, defect initiation	Very high to prohibitive

1. **SciML for multi-physics problems.** SciML architectures such as PINNs and NOs enable the simultaneous enforcement of multiple governing laws through residual terms or operator learning. Early demonstrations in MAM have shown that embedding thermo-fluid and thermo-mechanical coupling within the learning process improves the prediction of melt-pool geometry, temperature gradients, and stress development. These approaches highlight the potential for fast surrogate models that remain grounded in physical interactions.
2. **SciML for multi-scale problems.** To resolve steep gradients in the melt-pool region while retaining coherent part-scale behavior, several strategies have been proposed. These include partitioning the spatial domain into subregions trained with local surrogates, and enriching the spectral representation of neural networks to better capture high-frequency content. Such formulations mitigate the effects of spectral bias and improve accuracy in regions where resolution of fine features is essential for defect detection.
3. **SciML for time-dependent problems.** Recent studies incorporate temporal causality more explicitly, either by decomposing the time domain into sequential intervals or by assigning greater weight to earlier-time residuals to prevent error accumulation. NOs trained to propagate states forward in time have also enabled efficient long-horizon integration. These advances improve stability in dynamic regimes dominated by rapid melting and solidification.

Table 4 summarizes representative works aligned with these three modeling challenges and highlights their relevance to PBF physics, either directly or through transferable algorithmic advances. The subsections that follow provide a more detailed examination of these contributions in relation to each specific challenge.

SciML for multi-physics problems

A key requirement for high-fidelity PBF simulation is the simultaneous representation of several coupled physical fields, including heat transfer, melt-pool fluid flow, phase change, and mechanical response. In this subsection, we review SciML approaches that explicitly target such multi-physics coupling using either PINNs or NOs.

One of the earliest attempts at modeling multiphysics melt-pool behavior with PINNs was presented in Zhu et al. (2021). The powder layer was homogenized as a continuum, and the melt-pool surface was assumed flat to avoid free-surface tracking and vapor effects. To reduce the complexity of the loss function, the authors enforced Dirichlet boundary conditions as *hard constraints* and used FEM simulation data to guide training. This was implemented through an output transformation of the form:

$$u = u_{\text{bc}}[1 - H_\epsilon(x)] + u_{NN} H_\epsilon(x) \tag{28}$$

$$p = p_{\text{bc}}[1 - H_\epsilon(x)] + p_{NN} H_\epsilon(x) \tag{29}$$

$$T = T_{\text{bc}}[1 - H_\epsilon(x)] + T_{NN} H_\epsilon(x) \tag{30}$$

where u, p, and T denote the velocity, pressure, and temperature; $H_\epsilon(x) = 0$ for points on the boundary; and the subscript bc indicates prescribed conditions. While this work proved feasibility, the absence of gas dynamics, evaporation, and surface deformation limited predictive fidelity, and reliance on labeled FEM data prevented PINNs from functioning as a standalone surrogate.

A related thermo-mechanical study for DED was presented in (Sharma & Guo, 2025), where a dual-network architecture predicted temperature and resulting stresses. Here too, labeled FEM samples were required to stabilize training, highlighting the challenge of learning strongly coupled physics solely from governing equations.

To avoid dependency on supervised data, Harandi et al. (2024) adopted a *mixed-form* loss construction. Based on the principle that PINNs generally perform better when the loss

Table 4 Selected SciML publications categorized by addressed modeling challenges, with indication of the targeted AM process when applicable. Publications directly addressing AM are highlighted in gray, with darker shading indicating explicit demonstration on PBF.

Table 4: Selected SciML publications categorized by addressed modeling challenges, with indication of the targeted AM process when applicable. Publications directly addressing AM are highlighted in gray, with darker shading indicating explicit demonstration on PBF.

Challenge	Physics/AM Process	Method	References
Multi-physics	PBF	Thermo-fluid PINN for LPBF melt pool, with hard BCs and FEM support.	(Zhu et al., 2021)
Multi-physics	DED	Thermo-mechanical dual-network PINN supported by FEM labels.	(Sharma & Guo, 2025)
Multi-physics	Linear thermo-mechanic	Mixed-form PINN improving conditioning for coupled thermo-mechanics.	(Harandi et al., 2024)
Multi-physics	DED	DeepONet surrogate for final stress prediction from geometry + printing speed.	(Kushwaha et al., 2024)
Multi-physics	General / Electroconvection	DeepM&Mnet composite DeepONets for multi-physics coupling.	(Cai et al., 2021)
Multi-physics	General multiphysics	Coupled multi-FNO with latent RNN/attention exchange across physics.	(Li et al., 2025)
Multi-physics	General multiphysics	Hybrid kernel-inspired neural operator for nonlinear coupled PDEs.	(Yuan et al., 2025)
Multi-scale	General	cPINN domain decomposition enforcing flux continuity.	(Jagtap, Kharazmi, & Karniadakis, 2020)
Multi-scale	General	Parallel multi-GPU XPINN for large nonlinear domains.	(Shukla et al., 2021)
Multi-scale	General	FBPINN overlapping windows + normalization for multiscale.	(Moseley et al., 2023)
Multi-scale	General	APINN soft gating and theoretical generalization.	(Hu et al., 2023)
Multi-scale	General	Prototype-guided subdomain refinement via clustering.	(Wang et al., 2025)
Multi-scale	General	Fourier feature embeddings mitigate spectral bias.	(Wang, Wang, & Perdikaris, 2021)
Multi-scale	General	DFT-based Fourier warm start improves frequency learning.	(Jin et al., 2024)
Multi-scale	General	STaN locally adaptive activation for scale variation.	(Gnanasambandam et al., 2023)
Multi-scale	General multiphysics	Multipole GNO capturing multi-resolution physics (V-cycle).	(Li et al., 2020)
Time-dependent	General	PPINN: time-domain decomposition + solver-aided corrections.	(Meng et al., 2020)
Time-dependent	General	Parallelizable time-window PINNs with blending functions.	(Roy & Castonguay, 2024)
Time-dependent	General	Discrete-time PINNs with trapezoidal evolution.	(Jung et al., 2024)
Time-dependent	General	DC-Net enforcing conservation to prevent drift.	(Wandke & Zhang, 2024)
Time-dependent	General	Causality-weighted loss stabilizing long-horizon learning.	(Wang et al., 2024)
Time-dependent	General	bc-PINN sequential window retraining with boundary consistency.	(Mattey & Ghosh, 2022)
Time-dependent	General	DeepONet time-marching operator for long-term evolution.	(Wang & Perdikaris, 2023)

involves lower-order derivatives (Samaniego et al., 2020), the authors replaced strong-form residuals with variational principles where available. For mechanical equilibrium

$$\begin{aligned} \nabla \cdot \sigma &= 0 \quad \text{in } \Omega \\ u &= \bar{u} \quad \text{on } \Gamma_D \\ \sigma \cdot n &= \bar{t} \quad \text{on } \Gamma_N \end{aligned} \tag{31}$$

they minimized the elastic energy functional

$$\min \ \frac{1}{2} \int_{\Omega} \varepsilon C \varepsilon \, dV - \int_{\Gamma_N} u^T \bar{t} \, dA, \tag{32}$$

with $\varepsilon = \frac{1}{2}(\nabla u + \nabla u^T)$ and $\sigma = C\varepsilon$.

A similar variational form was used for steady heat conduction (Harandi et al., 2024):

$$\begin{aligned} \nabla \cdot (k \nabla T) &\quad \text{in } \Omega \\ T &= \bar{T} \quad \text{on } \Gamma_D \\ -k \nabla T \cdot n &= \bar{q} \quad \text{on } \Gamma_N \end{aligned} \tag{33}$$

$$\min \ \frac{1}{2} \int_{\Omega} k \, \nabla T^T \nabla T \, dV - \int_{\Gamma_N} \bar{q} \, T \, dA. \tag{34}$$

Coupled and sequential training strategies were compared, with the latter proving more efficient. However, these formulations apply only to linear physics; nonlinear melt-pool flow and phase change in PBF do not generally admit such energy functionals.

To improve flexibility and predictive generalization, recent works have leveraged NOs. These methods learn mappings between function spaces, enabling efficient parameterized predictions once trained. In the context of PBF process simulation, this means that NOs can, for example, learn a mapping between local material characteristics and the corresponding approximation of the temperature field.

In Kushwaha et al. (2024), a modified DeepONet architecture estimated thermal stresses at the end of a DED process. A ResUNet-based trunk network (He et al., 2023) encoded the part geometry from a binary mask, while a branch network elaborated process parameters such as the printing speed and additional structural embeddings from the trunk (see Figure 5). The operator learned by the model maps these inputs to the final stress state with accuracy comparable to FEM, at much lower computational cost.

Another strategy, DeepM&Mnet (Cai et al., 2021), decomposes the learning task by training multiple DeepONets separately, each representing a single physical variable. These pre-trained operators are then coupled either in parallel (for joint variable prediction) or in a serial fashion (where the prediction of one variable is used to reconstruct the others). This modular design makes the learning problem more tractable and facilitates scalable multi-physics coupling. In the context of PBF simulations, this means that different DeepONets could be assigned to distinct physical fields (such as temperature, velocity, and, in meso-scale models, pressure) while maintaining a coupled representation of the underlying physics.

A similar philosophy motivates the composite FNO introduced in Li et al. (2025). Here, M separate operators model distinct physics, while nonlinear layers exchange information via a global latent vector z_l:

$$z_l = \text{RNN}(z_{l-1}, V_l) \quad \text{or} \quad z_l = \text{Attention}(V_l),$$

where V_l concatenates intermediate representations from all operators. Such cross-physics aggregation improves learning of coupled behaviors.

Finally, Yuan et al. (2025) assumed that solutions of nonlinear multi-physics PDEs can be approximated as Urysohn-type integral operators:

$$u(x) = \int_{\Omega} k(x, x') f[x', a(x')] \, dx' + g(x), \tag{35}$$

and constructed a hybrid architecture combining a FNO (for learning the nonlocal operator kernel) with a convolutional network (for spatial feature extraction). This parallel design further improves efficiency for high-fidelity coupled systems.

Overall, these studies demonstrate that PINNs and NOs can embed multi-physics constraints directly into the learning process. Yet, most current applications remain constrained to simplified representations or require supervision from numerical solvers. Extending these approaches to fully predictive PBF simulation will therefore demand more expressive architectures and improved strategies for handling nonlinear physics and coupling strength.

SciML for multi-scale problems

As discussed in Section 3, PBF exhibits pronounced multiscale behavior: near the melt pool, geometry and fields vary over microns and microseconds, while at the part scale heat diffuses over centimeters and seconds. A single surrogate must therefore capture sharp, localized, high-frequency structures alongside smooth, global, low-frequency trends. In PINNs, this mismatch typically produces optimization stiffness and spectral bias; in NOs, it challenges the learned representation to reconcile heterogeneous local behavior

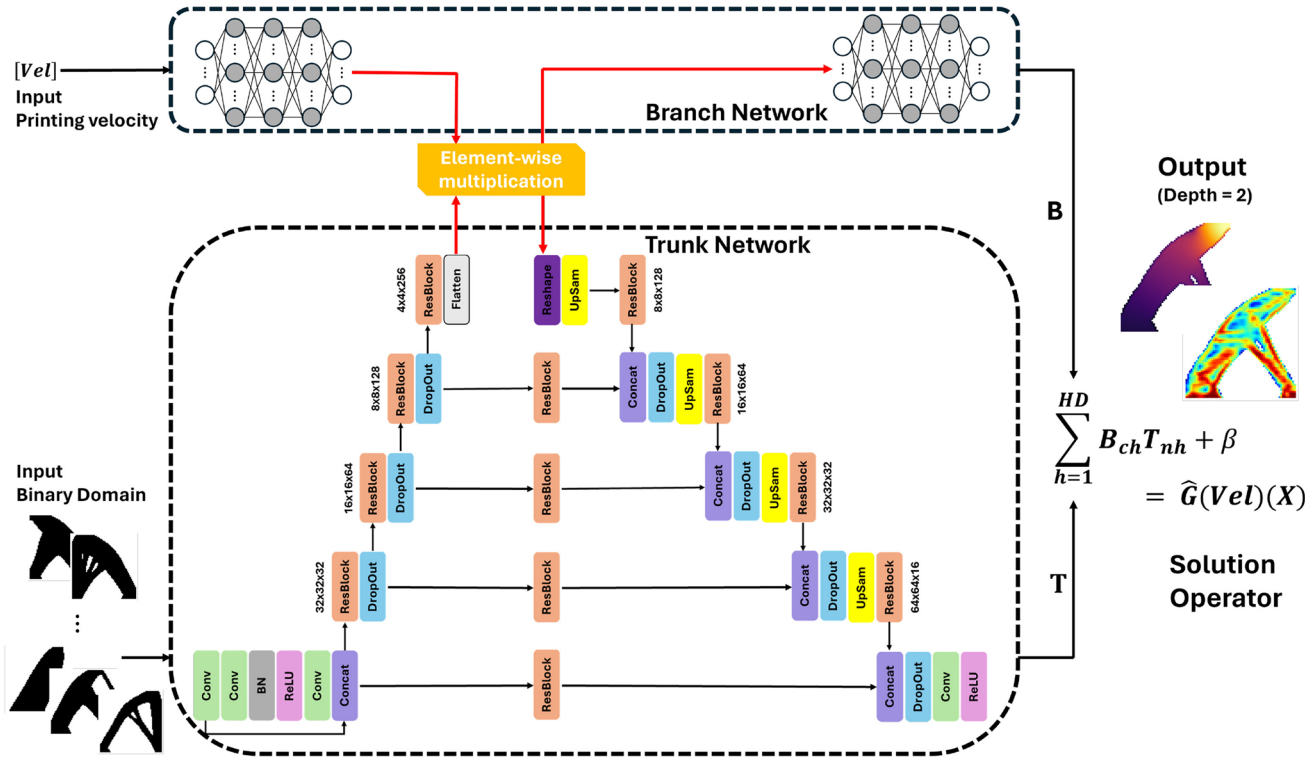


Fig. 5 Modified ResUNet-based DeepONet architecture for multiphysics surrogate modeling, where Vel is the printing speed, $\hat{G}$ is the learned solution operator, and HD is the dimensionality of the latent feature space. (adapted from (Kushwaha et al., 2024) under CC BY)

with long-range couplings. Two complementary research directions have emerged to address these issues: *(i)* domain decomposition, which lets models focus on local scales while enforcing global consistency, and *(ii)* spectral enrichment, which expands a network's ability to represent multiple frequency bands.

Domain decomposition approaches

Domain decomposition splits the computational domain into subdomains so that each network specializes locally, while interface conditions ensure global solution consistency. For the selection of subdomains in PBF simulations, there is no established rule; however, as in traditional numerical methods, it is generally more effective to use smaller subdomains in regions where field gradients are large (Jagtap et al., 2020b). In the context of PBF, this typically corresponds to finer subdomain partitioning near the melt pool, where thermal and fluid-dynamic gradients are strongest, and coarser partitions in the surrounding bulk material. A foundational contribution is the conservative physics-informed neural network (Conservative PINN (cPINN), Jagtap et al. (2020b)). Here, the domain is partitioned into multiple subdomains, each modeled by a dedicated network. The loss function includes interface penalties that enforce (i) flux continuity and (ii) agreement between neighboring solutions. Flux continuity is especially natural for conservation laws, where correct transmission across interfaces preserves physical fidelity.

The extended framework eXtended PINN (xPINN) (Jagtap & Karniadakis, 2020) generalizes this idea to more general PDEs. Rather than computing interface normals explicitly, xPINN enforces flux consistency by matching the residuals of the governing equations along interfaces. This simplification eases implementation for complex geometries and dynamics. Both cPINN and xPINN have been realized in parallel multi-GPU implementations (Shukla et al., 2021), enabling scalability for large domains and fine resolutions typical of multiscale problems.

A further step toward practicality is the finite-basis PINN (Finite-Basis PINN (FBPINN), Moseley et al. (2023)), which introduces overlapping subdomains blended by window functions and equipped with input/output normalization tailored to local scales. The global surrogate takes the form

$$u(x;\theta) = \sum_{i=1}^{n} w_i(x) \cdot \mathrm{unnorm}\big(NN_i(\mathrm{norm}_i(x);\theta)\big), \qquad (36)$$

where $w_i(x)$ is a window function that softly gates each subnetwork's contribution. A custom training schedule proceeds from outer to inner patches to propagate reliable information inward and accelerate convergence.

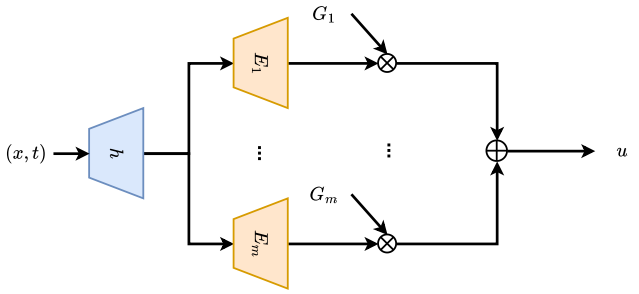


Fig. 6 Structure of APINN (Hu et al., 2023). A shared encoder extracts global features; gate networks $G_i(x)$ compute soft weights that blend m subnetworks $E_i(x)$, enabling adaptive domain decomposition for multiscale behavior

Beyond fixed partitions, adaptive schemes learn the decomposition. The Augmented PINN (APINN) architecture (Hu et al., 2023) uses a shared encoder $h(x)$ for global features and gate networks $G_i(x)$, either learned or specified as fixed windows, to produce soft assignments to m shared subnetworks $E_i(x)$ (see Figure 6). The authors also provide theoretical generalization bounds, supporting the method's stability.

A prototype-guided strategy in Wang et al. (2025) further refines adaptivity. After an initial training phase with an encoder $E(x)$ and m decoders $D_i(\cdot)$ constrained via

$$u(x) = \frac{1}{m}\sum_{i=1}^{m} D_i[E(x)],$$

k-means clustering on encoder outputs yields m prototypes Pro_i representing dominant local behaviors. During refinement, these prototypes define soft weights

$$w_i(x) = \frac{\exp\langle \mathrm{Pro}_i, E(x)\rangle}{\sum_{j=1}^{m}\exp\langle \mathrm{Pro}_j, E(x)\rangle}, \tag{37}$$

and the surrogate becomes

$$u(x) = \sum_{i=1}^{m} w_i(x) D_i[E(x)]. \tag{38}$$

A regularizer encourages diversity among prototypes (e.g., via orthogonality), improving coverage of disparate local scales. Empirically, this approach outperforms APINN on sharply localized features.

Spectral enrichment approaches

A second family of methods tackles spectral bias directly by enriching a network's effective frequency content. The study in Wang et al. (2021b) shows (empirically and, in simplified settings, theoretically) that random Fourier feature mappings (Tancik et al., 2020) expand the neural tangent kernel spectrum and improve multiscale accuracy. Inputs are lifted by sinusoidal embeddings

$$\begin{cases} \gamma^{(i)}(x) = \begin{bmatrix}\cos(2\pi B^{(i)}x)\\ \sin(2\pi B^{(i)}x)\end{bmatrix}, & i = 1,\dots,M,\\ H_1^{(i)} = \phi(W_1\cdot\gamma^{(i)} + b_1), & i = 1,\dots,M,\\ H_l^{(i)} = \phi(W_l\cdot H_{l-1}^{(i)} + b_l), & l = 2,\dots,L,\ i = 1,\dots,M,\\ u(x) = W_{L+1}\cdot\left[H_L^{(1)}\ \cdots\ H_L^{(M)}\right] + b_{L+1}, \end{cases} \tag{39}$$

with $(B^{(i)}) \sim \mathcal{N}(0_d, \sigma_i I_d)$ with $x \in \mathbb{R}^d$. By spanning a range of σ_i, the model becomes sensitive to both low- and high-frequency content, helping recover sharp gradients near melt-pool boundaries while retaining global coherence.

Building on this idea, Jin et al. (2024) proposes a data-informed "Fourier warm start" based on the discrete Fourier transform (DFT) of available signals,

$$F_j(\xi_c) = \sum_{k=1}^{n_j} u_{jk}\, \mathrm{e}^{-i\frac{2\pi}{n_j}ck}, \qquad c = 1,\dots,n_j, \tag{40}$$

with $\xi_c = \frac{2\pi}{n_j}c$. Let $a_{jc} = |F_j(\xi_c)|$ and $A_j = \sum_{c=1}^{n_j} a_{jc}$. The method allocates $N_{jc} = \lfloor m\, a_{jc}/A_j \rfloor$ random features to band $\mathcal{N}(\xi_c, \epsilon^2)$, favoring informative frequencies; a skip connection from Fourier features to the last hidden layer further smooths the loss landscape near minima.

Architectural choices also matter. The self-scalable tanh (stan) activation (Gnanasambandam et al., 2023),

$$\sigma_k^i(x) = (1 + \beta_k^i)\tanh(x), \tag{41}$$

introduces a trainable gain β_k^i that lets neurons adapt to local frequency content. This improves representation of regions with different characteristic scales, which is common when a single surrogate covers both melt-pool neighborhoods and cooler, slowly varying areas.

Finally, multipole-based NO provide a multiresolution mechanism akin to multigrid methods. The graph NO in Li et al. (2020) leverages fast multipole ideas (Greengard & Rokhlin, 1997) and multi-resolution matrix factorization (Kondor et al., 2014) to approximate nonlocal kernels at multiple scales. At each iteration,

$$v^{(t)}(x) = \sigma\Big[(W + \mathcal{K}_a)v^{(t-1)}(x)\Big]$$
$$\approx \sigma$$

$$\left[W v^{(t-1)}(x) + \frac{1}{|N(x)|} \sum_{y \in N(x)} k_\phi\big(a(x), a(y), x, y\big)\, v^{(t-1)}(y) \right], \tag{42}$$

where $N(x)$ are local neighborhoods and k_ϕ is a learned kernel. The full V-cycle connects coarse-to-fine and fine-to-coarse updates through restriction and prolongation:

$$\begin{cases} \check{v}_1^{(0)} = Pa, \\ \check{v}_{l+1}^{(t+1)} = \sigma\big(\hat{v}_{l+1}^{(t)} + \mathcal{K}_{l+1,l}\check{v}_l^{(t+1)}\big), \\ \quad l = 1, \ldots, L,\ t = 0, \ldots, T-1, \\ \hat{v}_l^{(t+1)} = \sigma\Big[\big(W_l + \mathcal{K}_{l,l}\big)\check{v}_l^{(t+1)} + \mathcal{K}_{l,l-1}\hat{v}_{l-1}^{(t+1)}\Big], \\ \quad l = L, \ldots, 1,\ t = 0, \ldots, T-1, \\ u = Q\hat{v}_1^{(T)}, \end{cases} \tag{43}$$

where P and Q are projection and reconstruction operators. Coarse levels propagate global information efficiently; fine levels recover localized details. Figure 7 shows an example of the structure of the kernels with 4 resolution levels. Variants with "W" and "F" cycles are explored in Migus et al. (2022).

In summary, *domain decomposition* helps surrogates assign resolution where needed and move information stably across interfaces, whereas *spectral enrichment* broadens the representable frequency content to capture both smooth and rapidly varying structures. Because PBF combines globally evolving thermal fields with highly localized melt-pool physics, combining these strategies is particularly promising for accurate and scalable SciML surrogates.

SciML for time-dependent problems

Time-dependent physics plays a central role in PBF, where the moving heat source induces rapid melting and solidification cycles, while global thermal diffusion evolves across much longer time scales. This results in stiff dynamical behavior, making standard PINNs prone to collapsing toward trivial or steady-state solutions when trained over long horizons. To address these challenges, two main classes of SciML techniques have been developed: (i) temporal domain decomposition methods (distinct from general domain decomposition approaches, as they explicitly account for the causal nature of the time variable by partitioning the temporal domain into sub-intervals), and (ii) loss formulations that enforce causality during training.

Domain decomposition across time

A representative method in the direction of temporal domain decomposition is the Parareal PINN (PPINN) introduced in Meng et al. (2020). The simulation interval $[0, T]$ is divided into N subdomains, and each sub-interval is modeled by a separate PINN $\mathcal{F}(u_i^k)$, where u_i^k denotes the network approximation at time t_i. A simplified conjugate-gradient-based solver $\mathcal{G}(u_i^k)$ provides a prediction that is iteratively corrected using the PINN output:

$$u_{i+1}^{k+1} = \mathcal{G}(u_i^{k+1}) + \delta_i^k = \mathcal{G}(u_i^{k+1}) + \mathcal{F}(u_i^k) - \mathcal{G}(u_i^k). \tag{44}$$

Training proceeds subdomain by subdomain until global convergence is achieved. This approach reduces the temporal span over which the network must remain accurate, alleviating long-term error accumulation.

A different formulation is proposed in Roy and Castonguay (2024), which also decomposes the interval into subdomains but avoids recursive dependency so that the networks can be trained in parallel. The solution is constructed as

$$\begin{cases} u_1(x, t; \theta_1) = \check{h}_1(t) g(x) + \hat{h}_1(t) f_1(x, t; \theta_1), \\ u_{n+1}(x, t; \theta_{n+1}) = \check{h}_{n+1}(t) f_n(x, t; \theta_n) \\ \quad + \hat{h}_{n+1}(t) f_{n+1}(x, t; \theta_{n+1}), \quad \forall n \geq 1, \end{cases} \tag{45}$$

where the blending functions satisfy $\check{h}_{n+1}|_{\tau=0} = \hat{h}_{n+1}|_{\tau=1} = 1$ and $\check{h}_{n+1}|_{\tau=1} = \hat{h}_{n+1}|_{\tau=0} = 0$, with $\tau = \frac{t-t_n}{t_{n+1}-t_n}$. This structure ensures continuity and efficient gradient propagation while enabling parallel training.

Another closely related strategy is introduced in Jung et al. (2024). Instead of assigning one model per interval, the authors discretize the time domain and define a distinct network $u(t_n, x, \theta_n)$ at each time level. For a generic equation $u_t + \mathcal{N}[u] = 0$, they approximate the temporal evolution using a trapezoidal discretization:

$$\begin{aligned} u(t_n, x) &= u(t_{n-1}, x) - \int_{t_{n-1}}^{t_n} \mathcal{N}[u](t, x)\, dt \\ &\approx u(t_{n-1}, x) - \frac{1}{2} h\big[\mathcal{N}[u](t_{n-1}, x) + \mathcal{N}[u](t_n, x)\big], \end{aligned} \tag{46}$$

with $h = T/N$. This eliminates the need for explicit time derivatives and enables parallel training across time slices, significantly improving efficiency.

Sequential training with additional constraints is also explored in Wandke and Zhang (2024), where a dedicated "deep conserved" network (Drift-Correcting network(DCnet)) is trained alongside each time-interval PINN. The DCnet enforces conservation properties (e.g., energy conservation) by ensuring that the conserved quantity at the start and end of each interval remains consistent, improving long-term stability.

Causality-aware temporal learning

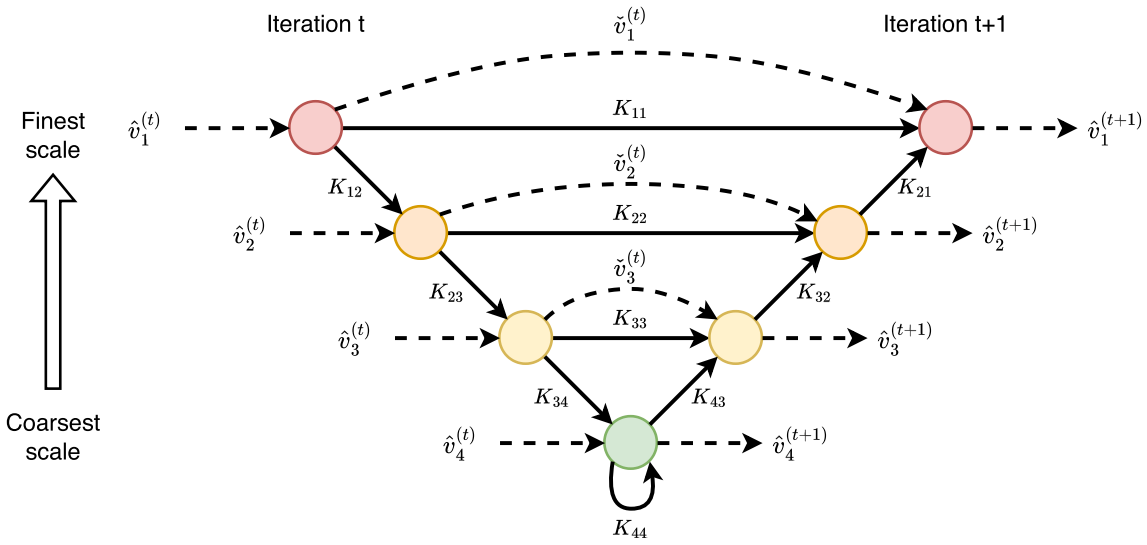


Fig. 7 Hierarchical V-cycle GNO architecture (Li et al., 2020). The figure provides a graphical representation of Equation (43) with $L = 4$ resolution levels arranged in a V-cycle structure. Coarse resolution levels efficiently capture long-range and global interactions, while progressively finer levels resolve localized, high-frequency features typical of melt-pool dynamics in PBF processes. Multiscale information is transmitted across levels through downward and upward passes of the V-cycle, enabling bidirectional exchange between coarse and fine representations. This hierarchical coupling is learned through the architectural bias introduced by scale-dependent kernel operators, allowing the model to simultaneously represent global context and local physical phenomena

An alternative to domain decomposition is to explicitly encode temporal causality. The causality-based PINN loss introduced in Wang et al. (2024) weights the residual contributions in time according to their chronological order:

$$\mathcal{L}(\theta) = \frac{1}{N_t} \sum_{i=1}^{N_t} w_i \, \mathcal{L}(t_i, \theta), \tag{47}$$

where $\mathcal{L}(t_i, \theta)$ is the physics loss at time t_i. The weights

$$w_i = \exp\left[-\epsilon \sum_{k=1}^{i-1} \mathcal{L}(t_k, \theta) \right] \tag{48}$$

promote accurate learning at earlier times before allowing influence from later states, counteracting the tendency to converge to temporally trivial solutions.

A related sequential approach is *bc*-PINN (Mattey & Ghosh, 2022), in which a single network is retrained for each sub-interval using a loss that enforces agreement with the previously computed solution at the boundary between intervals. This avoids the accumulation of time-dependent errors and results in more robust long-horizon predictions.

Finally, operator learning also supports time integration. In Wang and Perdikaris (2023), a physics-informed DeepONet is trained to map any initial condition to the solution over a short interval $[0, \Delta t]$. The operator is then composed sequentially so that the output at one stage becomes the initial condition for the next, analogous to a neural time-stepping scheme. This enables accurate long-term evolution while maintaining the mesh-free flexibility of NOs.

Overall, the time-dependent behavior of PBF introduces substantial training challenges for SciML models. Domain decomposition and causality-aware formulations represent two effective and complementary strategies to address stiffness and error accumulation over long time horizons, improving the stability and physical reliability of surrogate simulations.

Addressing the training issues of SciML methods

As discussed in Section 2, the numerical optimization problems underlying PINN- and NO-based surrogates remain a major bottleneck for their adoption in high-fidelity PBF modeling. In particular, loss imbalance, spectral bias, and stiff time integration can prevent convergence or lead to inaccurate, overly smoothed, or even trivial solutions. This section reviews methodological advances specifically designed to improve training stability, with emphasis on techniques most relevant to the multiphysics, multiscale, and transient behavior of PBF processes. We first examine strategies developed for PINNs, and then discuss related approaches for NOs.

Training stabilization methods for PINNs

In PINNs, the residual loss associated with the governing equations may dominate the boundary and initial condition losses due to their different gradient scales and collocation point distributions (Section 2). This imbalance induces stiffness in the optimization dynamics, preventing the model from accurately propagating constraints through the domain (Wang et al., 2021a). To mitigate this issue, three main categories of techniques have emerged: (i) adaptive loss balancing, (ii) collocation point weighting, and (iii) optimization constraints derived from statistical or geometric principles.

Adaptive loss balancing

One of the first works to quantify gradient imbalance introduced a *learning-rate annealing* mechanism that adjusts the loss weights λ_i according to the relative magnitude of the gradients (Wang et al., 2021a). Given

$$\mathcal{L}(\theta) = \mathcal{L}_r(\theta) + \sum_{i=1}^{M} \lambda_i \mathcal{L}_i(\theta), \tag{49}$$

the weights are computed as

$$\hat{\lambda}_i = \frac{\|\nabla \mathcal{L}_r(\theta)\|_\infty}{\frac{1}{|\theta|}\|\nabla \mathcal{L}_i(\theta)\|_1}, \tag{50}$$

smoothed via a moving average:

$$\lambda_i = (1-\alpha)\lambda_i + \alpha \hat{\lambda}_i. \tag{51}$$

A statistical refinement, *inverse Dirichlet weighting*, replaces the gradient norms with their standard deviations across parameters:

$$\hat{\lambda}_i = \frac{\max_j \operatorname{std}_\theta\{\nabla \mathcal{L}_j(\theta)\}}{\operatorname{std}_\theta\{\nabla \mathcal{L}_i(\theta)\}}, \tag{52}$$

yielding more balanced gradient distributions (Maddu et al., 2022).

A complementary physics-based solution exploits the NTK spectral theory (Wang et al., 2022). By aligning the eigenvalues of boundary and residual sub-kernels,

$$\lambda_b = \frac{\operatorname{Tr}(K)}{\operatorname{Tr}(K_{bb})}, \quad \lambda_r = \frac{\operatorname{Tr}(K)}{\operatorname{Tr}(K_{rr})}, \tag{53}$$

training is steered toward balanced learning of low- and high-frequency modes.

Other works formulate the problem geometrically, projecting parameter updates onto a descent direction that benefits all loss components:

$$g_{\text{ConFIG}} = \sum_{i=1}^{m} (g_i^T g_u) g_u, \tag{54}$$

where g_u ensures coherent optimization across tasks (Liu et al., 2025). Fractional-power loss rescaling has also been proposed to temporarily flatten the optimization landscape and progressively restore full residual penalization (Wang et al., 2024):

$$\mathcal{L}(\theta) = \omega_b \mathcal{L}_b^{1/n_b}(\theta) + \omega_r \mathcal{L}_r^{1/n_r}(\theta). \tag{55}$$

Adaptive collocation point weighting

Local importance weighting shifts attention to regions where the residuals or boundary errors are large. Self-Adaptive PINN (SA-PINN) (McClenny & Braga-Neto, 2023) introduces soft-attention masks:

$$\mathcal{L}_r(\theta, \lambda_r) = \frac{1}{2}\sum_{i=1}^{N_r} m(\lambda_{r,i}) |\mathcal{N}[u](x_i, t_i) - f(x_i, t_i)|^2, \tag{56}$$

$$\mathcal{L}_b(\theta, \lambda_b) = \frac{1}{2}\sum_{i=1}^{N_b} m(\lambda_{b,i}) |\mathcal{B}[u](x_i, t_i) - g(x_i, t_i)|^2, \tag{57}$$

$$\mathcal{L}_0(\theta, \lambda_0) = \frac{1}{2}\sum_{i=1}^{N_0} m(\lambda_{0,i}) |u(x_i, 0) - u_0(x_i)|^2, \tag{58}$$

and updates weights via gradient ascent.

Variants include domain-adaptive residual weighting (Differentiable Adversarial SA-PINN (DASA-PINN), (Zhang et al., 2023)) and multi-network strategies for multiphysics (Song et al., 2024). To reduce computational cost, (Anagnostopoulos et al., 2024) introduces a deterministic update law:

$$\lambda_i^{(k+1)} = \gamma \lambda_i^{(k)} + \eta^* \frac{|r_i|}{\max_i |r_i|}. \tag{59}$$

Residual smoothing through convolution operators has been explored via min–max optimization (Si & Yan, 2025),

$$\min_\theta \max_{\lambda_r, \lambda_b, \lambda_i} \lambda_r^T r(\theta) - \frac{1}{2}\lambda^T \lambda + \lambda_b^T r_b(\theta) + \lambda_i^T r_i(\theta), \tag{60}$$

and its smoothed variant,

$$\min_{\theta} \max_{\lambda_r,\lambda_b,\lambda_i} \lambda_r^T W^{1/2} r(\theta) - \frac{1}{2}\lambda^T\lambda + \lambda_b^T r_b(\theta) + \lambda_i^T r_i(\theta), \quad (61)$$

where W encodes local residual correlations.

Finally, probabilistic loss weighting (Cao et al., 2025) treats each residual as a Gaussian variable with learnable variance, naturally promoting balanced learning.

Training stabilization methods for NOs

As reviewed in Section 2, NOs shift the learning task from solving a single realization of a PDE to learning the entire operator mapping between function spaces. This generality gives NOs a major advantage for parameterized PBF applications but introduces new training challenges linked to: (i) data scalability, (ii) discretization mismatch, and (iii) optimization instability in stiff multiphysics systems. Below, we categorize stabilization strategies that address these issues and are most relevant to PBF modeling.

Physics-constrained operator learning

Physics-informed NOs incorporate governing equations directly into the loss function. A generic formulation writes the operator training objective as

$$\begin{aligned}\mathcal{L}(\theta) = &\underbrace{\frac{1}{N_d}\sum_{j=1}^{N_d} \|\mathcal{G}_\theta[a_j] - u_j\|^2}_{\text{supervised term}} \\ &+ \lambda_r \underbrace{\frac{1}{N_r}\sum_{i=1}^{N_r} \|\mathcal{N}[\mathcal{G}_\theta(a)](x_i) - f(x_i)\|^2}_{\text{physics residual}} \\ &+ \lambda_b \underbrace{\frac{1}{N_b}\sum_{i=1}^{N_b} \|\mathcal{B}[\mathcal{G}_\theta(a)](x_i) - g(x_i)\|^2}_{\text{boundary loss}}\end{aligned} \quad (62)$$

mirroring the structure of (2), but without explicitly differentiating through the network outputs to enforce the dynamics at each collocation point. This improves the conditioning of the optimization problem, particularly when the underlying physics introduce stiffness (Cheng et al., 2025).

Boundary-aware architectures further reduce residual penalties by applying hard-constraint transformations, for example:

$$\tilde{u}(x) = u_{\text{bc}}(x) + \phi(x)\,\mathcal{G}_\theta[a](x), \quad (63)$$

with $\phi(x) = 0$ on $\partial\Omega$, similar to the classical PINN strategy in (6). Such formulations eliminate the boundary-loss imbalance, significantly improving stability on PBF-like problems involving localized heating.

Spectral regularization and mode-aware training

Fourier-based NOs, such as the FNO, compute updates of the form

$$v_{t+1}(x) = W v_t(x) + \mathcal{F}^{-1}\big[R_\phi(\xi)\cdot\mathcal{F}(v_t)(\xi)\big], \quad (64)$$

where R_ϕ learns to filter spatial frequency components ξ of the evolving latent representation. In multiphysics high-frequency regimes, naïvely learning all modes simultaneously often results in instability or divergence.

To mitigate this effect, (George et al., 2024) introduce *incremental spectral training*, gradually activating higher-frequency modes:

$$R_\phi(\xi) = \begin{cases} \hat{R}_\phi(\xi) & \text{if } \|\xi\| \le \omega(t), \\ 0 & \text{otherwise,} \end{cases} \quad \text{with } \omega(t) \text{ increasing over epochs.} \quad (65)$$

This strategy improves convergence by aligning optimization dynamics with the natural scale separation of the operator, which is crucial for PBF where melt-pool physics produce localized high-frequency features.

Complementary approaches impose spectral smoothness penalties (e.g., Tikhonov regularization in Fourier space), helping prevent unphysical oscillations in stress or temperature fields.

Discretization-invariant learning and multi-resolution supervision

A unique challenge in operator learning is the so-called *mesh transferability*: the operator should generalize across different discretizations of Ω. However, inference on a finer grid than that used during training can yield large errors, known as *discretization mismatch* (Gao et al., 2025).

To address this, multi-resolution training supervises both coarse and fine discretizations:

$$\mathcal{L}_{\text{MR}} = \sum_{\ell=1}^{L} \alpha_\ell \,\|\mathcal{G}_\theta[a^\ell] - u^\ell\|^2, \quad (66)$$

where ℓ indexes resolutions. This improves operator invariance to sampling patterns, a feature particularly beneficial when PBF geometries vary widely.

Graph-based operator architectures such as in (Hao et al., 2023) further enable learning on irregular or evolving domains, allowing modeling of powder-bed dynamics with complex free surfaces.

Hybrid supervision for stiff multiphysics dynamics

Physics-only supervision may be insufficient to guide operator training in regions where the governing equations become nearly singular (e.g., near solid-liquid boundary). Hybrid learning strategies combine labeled high-fidelity fields with physics residuals:

$$\mathcal{L} = \beta \|\mathcal{G}_\theta[a] - u_{\mathrm{HF}}\|^2 + (1-\beta)\,\|\mathcal{N}[\mathcal{G}_\theta(a)] - f\|^2, \qquad \beta \in [0,1], \tag{67}$$

ensuring stability without dependence on dense supervised data. Such strategies are especially aligned with PBF workflows, where high-fidelity samples can be triggered only when uncertainty exceeds process-specific thresholds.

A structured overview of these stabilization techniques and their relevance to PBF simulations is provided in Table 5.

Practical deployment of SciML methods: the current landscape

While the previous sections have demonstrated the theoretical capabilities of PINNs and NOs for modeling the thermo-fluid and thermo-mechanical behavior of MAM, with special regards to the PBF process, the transition from research prototypes to industrial deployment remains a significant challenge. Real manufacturing environments introduce constraints that extend beyond prediction accuracy, including software usability, real-time performance, seamless interaction with sensor data, and integration within digital twin architectures for process monitoring and control. This section reviews the current ecosystem of scientific machine learning software, examines representative applications in engineering practice, and evaluates the readiness of these technologies for enabling autonomous and reliable MAM workflows.

Available software frameworks

The implementation of PINNs and NOs in MAM applications depends not only on algorithmic advances, but also on the availability of software tools capable of handling complex multi-physics behavior while remaining accessible to practitioners with diverse technical backgrounds. Unlike conventional finite-element platforms, where mature graphical interfaces and streamlined workflows allow users to construct complex simulations without interacting directly with numerical solvers, current SciML frameworks still rely heavily on explicit coding of differential operators, sampling strategies, and loss formulations. Consequently, users must possess expertise in both programming and mathematical modeling, which poses a non-negligible barrier to adoption for process engineers and materials scientists who are the primary stakeholders in industrial AM environments. This gap in usability significantly slows the translation of emerging SciML methods into production-ready engineering tools.

All currently available software frameworks for SciML share three fundamental components. First, they provide tools for defining the geometry of the computational domain and sampling points within the interior and along the boundaries. Second, they offer interfaces for specifying the mathematical model, such as the governing differential operators and physical constraints, so that residuals can be evaluated during training. Third, they rely on a deep learning backend to define the neural network architecture and optimize its parameters. This general workflow is illustrated in Figure 8.

The neural network component is typically built on automatic differentiation libraries that compute both the gradients required for loss minimization and the derivatives of the network output necessary to evaluate the PDE residuals and boundary conditions. Commonly used backends include TensorFlow (Abadi et al., 2015), PyTorch (Paszke et al., 2019), JAX (Frostig et al., 2018), and PaddlePaddle (Baidu, 2016). Table 6 summarizes representative software libraries that integrate these capabilities for the development of PINNs and NOs, which are further discussed in the following paragraphs.

DeepXDE

DeepXDE (Lu et al., 2021b) is among the earliest and most widely adopted Python libraries for PINNs and operator learning. It supports a diverse range of formulations, including standard PINNs, residual-based adaptive refinement, energy-based and gradient-enhanced variants (Yu et al., 2022), stochastic PINNs (Zhang et al., 2020), and Deep Operator Networks (DeepONets). Its flexible API enables the specification of complex geometries and boundary conditions through constructive solid geometry, facilitating applications across fluid mechanics, geophysics, and biomechanics, such as tsunami wave propagation (Mao et al., 2020), multiphase flow in porous media (Jagtap et al., 2020a), and coronary artery hemodynamics (Kissas et al., 2020).

From a usability perspective, DeepXDE provides built-in tools for computing network derivatives and standard differential operators via automatic differentiation. However, users must explicitly define the governing equations, and the framework offers limited automation for hyperparameter tuning or model design. As such, effective use still requires substantial prior knowledge of the underlying physics and numerical formulation.

Table 5 Taxonomy of stabilization strategies for SciML surrogates relevant to PBF multiphysics modeling.

(a) Physics-Informed Neural Networks (PINNs)			
Method	**Key Idea**	**Relevance to PBF**	**References**
Adaptive Loss Balancing	Balance residual vs. boundary/initial terms to mitigate gradient stiffness. Techniques: dynamic weight updates, NTK eigenvalue alignment.	Improves enforcement of boundary conditions and solid–liquid phase transitions in multiphysics loss landscapes.	Learning-rate annealing (Wang et al., 2021a); Inverse Dirichlet weighting (Maddu et al., 2022); NTK-based scaling (Wang et al., 2022); ConFIG (Liu et al., 2025); Fractional loss rescaling (Wang et al., 2024).
Collocation Point Weighting	Pointwise importance weighting based on residual magnitude or learned attention.	Enhances accuracy near melt-pool boundaries, spatter trajectories, and lack-of-fusion defect zones.	SA-PINN (McClenny & Braga-Neto, 2023); DASA-PINN (Zhang et al., 2023); Component-wise weighting (Song et al., 2024); Bounded-weight law (Anagnostopoulos et al., 2024); Convolution min–max weighting (Si & Yan, 2025); Residual variance weighting (Cao et al., 2025).
(b) Neural Operators (NOs)			
Method	**Key Idea**	**Relevance to PBF**	**Examples**
Physics-Constrained Operator Learning	Add PDE residuals or hard-BC transformations to the operator mapping loss.	Improves multiphysics coupling without heavy reliance on labeled data.	Physics-informed NOs (Cheng et al., 2025).
Spectral Stabilization	Gradual activation or regularization of high-frequency Fourier modes.	Necessary for resolving steep thermal gradients and recoil-pressure-driven melt-pool oscillations.	Incremental spectral training (George et al., 2024); Spectral smoothing penalties.
Discretization Robustness	Enable mesh-transferable learning via multi-resolution objectives or grid-agnostic operators.	Supports rapid design changes in part geometry, hatch pattern, and layer thickness.	Multi-resolution supervision (Gao et al., 2025); Graph-based NO (gNO) (Hao et al., 2023).
Hybrid Supervision	Blend sparse high-fidelity data with physics priors to stabilize stiff regimes.	Mitigates uncertainty where vaporization, keyholing, and Marangoni flow dominate.	HF-guided updates in stiff transitions (e.g., melt-pool topology changes).

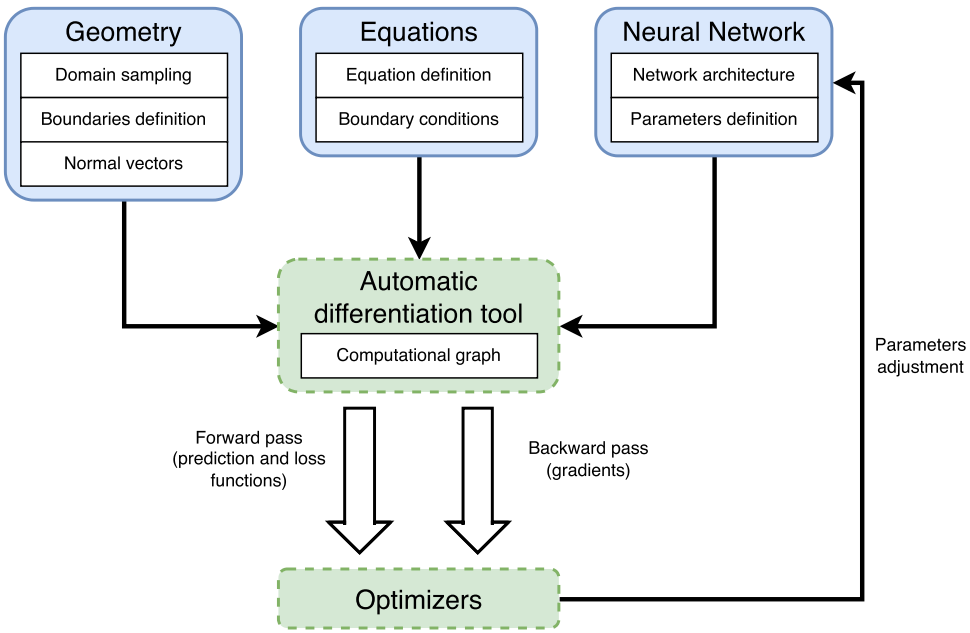


Fig. 8 General structure of a software framework for training SciML models. The workflow comprises the definition of the domain and physical model, sampling of training points or fields, a neural network that approximates either the solution or the operator, and gradient-based optimization enabled by automatic differentiation

Table 6 Overview of representative SciML libraries.

Software	Backend(s)	Language	Summary of characteristics
DeepXDE (Lu et al., 2021b)	PyTorch, TensorFlow, JAX, PaddlePaddle	Python	Broad algorithmic support; explicit PDE formulation required.
NeuralPDE.jl (Zubov et al., 2021)	Julia	Julia	Symbolic PDE definition; direct coupling with mechanistic solvers; best for expert users.
PINA (Coscia et al., 2023)	PyTorch	Python	Modular framework supporting PINNs and NOs; user-friendly syntax; suitable for research prototyping.
PhysicsNeMo (Contributors, 2025)	PyTorch	Python	Comprehensive physics-ML framework with predefined physics nodes (only for limited applications); GPU acceleration.

NeuralPDE.jl

NeuralPDE.jl (Zubov et al., 2021) is part of the Julia SciML ecosystem and enables symbolic specification of PDEs through a unified mathematical representation. It supports a variety of PINN formulations, including quadrature-based approaches (Kharazmi et al., 2019), and connects with NeuralOperators.jl, which implements operator-learning architectures such as DeepONet, the FNO, GNO, and Physics-Informed Neural Operator (PINO) (Li et al., 2024).

A notable advantage of NeuralPDE.jl is its seamless integration with DifferentialEquations.jl, which allows hybrid mechanistic-data-driven workflows where PINNs or NOs interact directly with high-fidelity ODE/PDE solvers. This functionality has enabled applications across multiple domains, including electrochemical battery modeling using the Doyle-Fuller-Newman equations and stochastic differential equations in finance.

Despite its extensive modeling flexibility, effective use of NeuralPDE.jl typically requires a strong understanding of the governing physics and numerical methods. Furthermore, the Julia ecosystem, while growing rapidly, remains less widely adopted than Python, which may limit accessibility for practitioners less familiar with scientific programming. Nonetheless, NeuralPDE.jl offers rich documentation and example-driven resources that support advanced research and experimentation.

PINA

PINA (Coscia et al., 2023) is a PyTorch-based library that unifies PINNs and neural operator models within a modular architecture. It supports standard and variational PINNs as well as operator-learning frameworks including DeepONet, the FNO, and the GNO. Built on PyTorch Lightning, PINA adopts a "Problem-Solver-Trainer" abstraction that simplifies scalable experimentation, distributed training, and rapid prototyping of multi-component models.

Applications demonstrated in the literature include benchmark PDEs such as the Helmholtz and diffusion equations and reduced-order surrogates for fluid dynamics (Coscia et al., 2023). From a usability standpoint, PINA provides a more user-friendly syntax than many early SciML frameworks and offers pre-implemented differential operators to reduce manual coding effort. However, effective use still requires explicit formulation of the governing physics and careful configuration of the problem setup, which may pose challenges for non-expert users.

PhysicsNeMo

PhysicsNeMo (formerly NVIDIA Modulus) (Contributors, 2025) is an open-source PyTorch framework optimized for GPU-accelerated scientific machine learning. It provides a model zoo of state-of-the-art architectures, including standard and variational PINNs, PINNs with adaptive activations (Jagtap et al., 2020a), Deep Operator Networks, the FNO, Adaptive FNO (Pathak et al., 2022), and hybrid PINO. These models have been applied to a wide range of engineering and geoscientific problems, from FNO-based Darcy flow simulations to global weather forecasting and lid-driven cavity flow prediction.

The framework uses a node-based computational graph in which each node can represent a variable, neural network component, differential operator, or manually defined algebraic expression. This structure enables efficient assembly of PDE operators and data constraints, while predefined physics modules help reduce the effort required to implement common governing equations. Additional features include utilities for importing external datasets, exporting predictions in multiple formats, and monitoring training performance–including per-loss diagnostics–via TensorBoard (Abadi et al., 2015).

Although PhysicsNeMo is currently the most feature-complete solution for SciML-based digital engineering applications, its automation and abstraction remain limited to the

physics domains supported by built-in operators. Extending the framework to new manufacturing scenarios still requires expert knowledge of the governing physics and significant customization.

Integration into Digital Twins for PBF

The integration of SciML into metal AM is driven by the need for fast, physics-consistent surrogates capable of supporting real-time monitoring, optimization, and control within a digital-twin framework. In an ideal PBF scenario, a digital twin continuously acquires in-situ sensor data, updates its internal prediction of temperature and melt-pool evolution, and autonomously adjusts process parameters to guarantee part quality, process stability, and repeatability. Achieving such capabilities represents a key requirement for next-generation manufacturing systems aligned with Industry 4.0 (Kizhakkinan et al., 2023).

Progress in other engineering domains provides encouraging evidence that SciML surrogates can enable functionalities directly relevant to digital-twin development. In aerospace, neural operators have accelerated aerodynamic shape optimization under PDE constraints (Shukla et al., 2024), demonstrating reactive design-space exploration. In the built environment, PINN-enhanced digital twins have supported predictive energy and autonomous control (Kazemi Naeini et al., 2025). At industrial scale, SciML surrogates have been applied to thermo-fluid optimization (Eidell et al., 2021; Susulovska et al., 2024), weather forecasting (Bonev et al., 2023), and power-grid simulation (Mathew et al., 2024), illustrating their scalability, real-time performance, and operational robustness. These precedents suggest that the key components of a digital twin-data assimilation, optimization coupling, and low-latency inference are technically achievable and can serve as a roadmap for analogous integration in PBF processes.

Current status and technical challenges

Despite encouraging developments in other sectors, the use of PINNs and NOs in MAM remains largely confined to *forward-modeling* tasks, in which neural networks replace high-fidelity numerical solvers to approximate thermo-fluid or thermo-mechanical fields (Yan et al., 2024). These studies demonstrate notable accuracy and computational efficiency but typically operate as offline surrogates, decoupled from equipment control and without mechanisms for uncertainty quantification or real-time feedback.

Recent academic efforts have begun exploring online control and parameter optimization, such as laser-power adjustment via PINNs (Depaoli et al., 2024) or surface-roughness regulation using NO (Liu et al., 2024). While these examples confirm the potential of physics-informed learning for process development, their deployment remains restricted to virtual testbeds with simplified sensing and control architectures.

Embedding SciML surrogates into cyber-physical production systems presents several unresolved technical barriers:

- **Low-latency, high-fidelity inference:** Surrogates must predict temperature, melt-pool geometry, and phase evolution at millisecond timescales to enable responsive control.
- **Robustness and generalization:** Reliable deployment requires uncertainty quantification to detect and handle out-of-distribution conditions.
- **Physics-constrained optimization:** Neural surrogates, physical models, and control algorithms must be tightly coupled to ensure feasible and stable corrective actions.
- **Sensor fusion and synchronization:** Digital twins must assimilate data from high-bandwidth modalities such as pyrometry, thermography, and melt-pool imaging (Di Cataldo et al., 2021) to maintain consistency with evolving machine conditions.

Technology-readiness assessment

To contextualize the current maturity of SciML components within PBF, their indicative Technology-Readiness Level (TRLs) were estimated based on representative studies in the literature. The assessment considers demonstrated capabilities from recent experimental and simulation-based works, including offline thermal surrogates (Pieressa et al., 2025; Yan et al., 2024), parametrized modeling efforts (Depaoli et al., 2024; Liu et al., 2024), and integration-oriented frameworks such as PhysicsNeMo (Contributors, 2025). Table 7 summarizes the corresponding readiness levels following the NASA and ESA TRL scales commonly adopted in manufacturing research (Mankins, 1995; Technology readiness levels handbook for space applications, 2017).

Although SciML surrogates have demonstrated promising predictive capabilities, significant gaps remain before full cyber-physical integration is achieved. As summarized in Table 7, current implementations span technology readiness levels (TRLs) between 2 and 5 depending on the targeted capability, highlighting several critical research directions required to increase maturity and enable industrial deployment.

Automation and accessibility: The moderate TRL (4–5) achieved by forward thermal surrogates in Table 7 is largely confined to controlled, offline studies. Advancing beyond this stage requires improved automation of model setup, including automated PDE specification, physics-aware loss construction, and hyperparameter tuning. Template-based workflows for common PBF sub-problems could reduce manual implementation effort and improve robustness when

Table 7 Indicative technology readiness levels (TRLs) of SciML components for PBF digital twins, estimated from representative studies (Contributors, 2025; Depaoli et al., 2024; Liu et al., 2024; Yan et al., 2024). TRL definitions follow the NASA/ESA scales (Mankins, 1995; Technology readiness levels handbook for space applications, 2017).

Capability	TRL	Status and limitations
Forward thermal surrogate (offline)	4-5	Reliable prediction in controlled studies; limited robustness to variations.
Multi-physics surrogate (thermal + mechanics + melt pool)	3-4	Accuracy challenges for tightly coupled phenomena; stability concerns.
Real-time simulation acceleration	2-3	Achieved only with simplified or reduced-order physics.
Closed-loop monitoring and control integration	2-3	Proof-of-concept only; no industrial deployment reported.

models are transferred across machines, materials, or process windows.

Multi-physics generalization: As indicated by the TRL 3–4 assigned to multi-physics surrogates in Table 7, tightly coupled thermal, fluid, and mechanical phenomena remain challenging to model reliably. Addressing the reported accuracy and stability limitations requires modular multi-physics architectures, shared latent representations across physical domains, and constraint-based regularization strategies that explicitly enforce conservation laws and coupling conditions.

Real-time deployment: The low TRL (2–3) reported in Table 7 for real-time simulation acceleration reflects the fact that real-time capability has only been demonstrated using simplified or reduced-order physics. Progress toward higher TRLs will depend on systematic model compression techniques, such as pruning, quantization, and low-rank representations, as well as hybrid approaches combining SciML surrogates with classical reduced-order models. Adaptive spatial-temporal resolution strategies focusing computational effort on critical regions, together with deployment on GPU-accelerated or edge hardware, are key enablers for latency-constrained operation.

Continual learning and adaptation: Across all capabilities listed in Table 7, robustness to process variability remains a limiting factor. To mitigate performance degradation caused by material variability, environmental changes, or equipment wear, SciML models must incorporate continual learning and domain adaptation strategies. Incremental updating schemes, coupled with regularization mechanisms that prevent catastrophic forgetting while preserving physical consistency, are essential for sustained performance in operational settings.

Verification, validation, and reliability: Finally, the TRL 2–3 assigned to closed-loop monitoring and control integration in Table 7 underscores the absence of validated industrial deployments. Raising the maturity of this capability requires standardized verification and validation frameworks, uncertainty quantification suitable for control-oriented decision making, and traceability mechanisms linking predictions to physical assumptions and data sources. Such developments are critical to meeting qualification and safety requirements in closed-loop manufacturing environments.

Addressing these priorities will transform current SciML surrogates from offline predictors into trusted, self-updating digital companions capable of monitoring, predicting, and controlling the PBF process in real time. This progression represents a critical step toward the realization of autonomous and reliable metal AM systems.

Conclusions and outlook

As emerged from our review, in the past few years SciML has evolved into a promising paradigm to bridge physics-based modeling and data-driven learning for metal additive manufacturing. In particular, PINNs and NOs have shown the potential to encode the governing physical laws of the PBF process while learning directly from data. However, the field remains in a transitional phase: while theoretical understanding and proof-of-concept demonstrations have advanced considerably, both the algorithms and their deployment pathways are still evolving toward practical maturity. Current limitations can be broadly categorized into two classes: *methodological* and *deployment-related*. The following discussion summarizes these challenges and highlights key directions toward reliable, real-time, and autonomous PBF manufacturing.

Methodological limitations and research directions

Although SciML methods have expanded rapidly in scope, several unresolved issues still constrain their predictive accuracy, robustness, and generality. From an industrial perspective, the urgency of these research directions is not uniform. The most immediate need concerns robust multiphysics representations, stable training dynamics, and multiscale coupling, as these directly affect predictive accuracy and reliability in operational PBF settings and they limit defect prediction and process-window generalization. Finally, uncertainty quantification, while essential for certi-

fication and qualification, currently depends on progress in the former directions and therefore represents a medium-term priority.

Complex multiphysics representation

To this date, only a limited subset of physical phenomena governing the PBF process has been modeled comprehensively. Most studies focus on large-scale heat transfer or residual stress formation, whereas melt-pool fluid dynamics, vaporization, and powder-scale interactions remain largely unexplored. Accurate modeling of these effects would require relaxing continuum assumptions and accounting for transient variations in material properties such as density and composition. Very few attempts have incorporated explicit particle-level dynamics (Wessels et al., 2020). Increasing the complexity of the models enables the development of tools capable of finer selection and control of process parameters, allowing a broader range of defects to be addressed. However, despite these potential benefits, the introduction of complex mathematical models reduces the effectiveness and practicality of SciML algorithms, particularly in terms of training stability and computational efficiency.

Training dynamics and stability

Balancing multiple loss components, mitigating spectral bias, and stabilizing stiff time-dependent systems remain open research problems. Existing weighting strategies often depend on instantaneous loss magnitudes or gradient norms, which do not necessarily correlate with global solution quality. Emerging analyses based on the Neural Tangent Kernel and network-state metrics (Anagnostopoulos et al., 2026) offer diagnostic insights but have yet to yield robust, general-purpose training protocols. As discussed above, improving existing algorithms or developing new ones is essential for accurately modeling multiple interacting physical phenomena.

Temporal and spatial multiscale coupling

The disparity between microsecond-scale melt-pool dynamics and macro-scale thermal diffusion leads to severe stiffness and gradient imbalance during training. Causality enforcement and domain decomposition alleviate these issues only partially (Rohrhofer et al., 2023), and the need for adaptive spatio-temporal resolution remains pressing. Architectures capable of dynamically allocating modeling capacity across scales are essential for consistent fidelity in PBF simulations.

Reliability and uncertainty quantification

For industrial and safety-critical applications, SciML surrogates must provide not only accurate predictions but also calibrated estimates of their uncertainty. Probabilistic physics-informed networks and Bayesian operator-learning formulations offer promising routes toward quantifiable confidence bounds and trustworthy deployment. Developing unified frameworks for uncertainty quantification and validation will be crucial to align SciML methods with industrial qualification standards.

Deployment and integration challenges

Alongside the methodological challenges, the practical integration of SciML into industrial AM workflows encounters equally significant obstacles. The current *software infrastructure* supporting SciML research remains largely prototypical in nature. Contemporary frameworks such as DeepXDE, NeuralPDE.jl, PINA, and PhysicsNeMo offer flexible environments for domain definition, differential operator specification, and neural surrogate training. However, effective use of these libraries requires substantial prior knowledge of the underlying algorithms and a clear understanding of their regimes of applicability to address the intrinsic challenges of SciML approaches. Moreover, these frameworks rely on explicit coding of governing equations, loss formulations, and hyperparameter choices, which further limits accessibility for engineers and materials scientists. In addition, existing libraries typically stop at the model development stage and do not support the direct deployment of trained models within industrial control or monitoring systems. To approach the usability and maturity of established finite-element platforms, future SciML frameworks must therefore provide automated PDE assembly, pre-configured multi-physics templates, and seamless pathways for deployment, together with tighter integration into commercial CAE environments and control ecosystems. Standardized datasets, benchmarking protocols, and open repositories also remain essential to ensure reproducibility and comparability across studies.

Beyond usability concerns, SciML surrogates must evolve from offline predictive tools into components capable of operating within *cyber-physical loops*. As summarized in Table 7, current implementations of real-time and closed-loop functionality typically remain at technology-readiness levels between TRL 2 and 3. Progress toward higher maturity will require advances in model compression, low-latency inference on GPU-accelerated or edge hardware, and continual-learning strategies that enable adaptation to process drift arising from material variability and equipment degradation.

Toward digital-twin-enabled additive manufacturing

As anticipated in Section 1, the long-term vision for SciML in metal additive manufacturing lies in the development of digital twins capable of real-time monitoring, reasoning, and control. In these systems, sensor data from pyrometry, thermography, or melt-pool imaging (Di Cataldo et al., 2021) would be assimilated continuously to update model predictions and autonomously adjust process parameters. Recent demonstrations, including the NO-based surface-roughness controller proposed in Liu et al. (2024) and the PINN-driven

parameter optimization in Depaoli et al. (2024), illustrate the feasibility of this approach in virtual testbeds. Transferring such frameworks to physical machines will require computational efficiency, robust uncertainty management, and standardized communication interfaces between models and control hardware. When these capabilities converge, real-time SciML surrogates embedded in digital twins could enable predictive monitoring, adaptive optimization, and self-correcting feedback mechanisms across the entire build process (Qi et al., 2021).

Concluding perspective

This survey evaluated the current maturity of SciML algorithms for process modeling and system-level integration in metal additive manufacturing, with particular emphasis on their role within emerging digital twin (DT) paradigms. Two main classes of challenges were identified: methodological limitations and deployment-related constraints. From a methodological perspective, existing SciML approaches have not yet reached full maturity and continue to struggle with the intrinsic complexity, strong multiphysics coupling, and multiscale nature of realistic PBF processes. In particular, the generalization capability of current models remains limited under extreme or out-of-distribution operating conditions, and practical upper bounds on achievable accuracy emerge when tightly coupled thermal, fluid, and mechanical phenomena are modeled simultaneously. As a result, SciML models trained without systematic support from high-fidelity numerical simulations or carefully curated experimental data often exhibit reduced predictive reliability.

From a deployment standpoint, current SciML software frameworks remain difficult to use for non-expert practitioners, and industrial MAM platforms provide limited support for the integration, validation, and long-term maintenance of externally developed models. These limitations hinder the direct adoption of SciML solutions within operational environments and constrain their current role within industrial digital twins, which are still predominantly limited to offline analysis or decision-support functionalities. Moreover, the lack of standardized benchmarks, qualification procedures, and uncertainty quantification strategies further complicates the assessment of model robustness and transferability across machines, materials, and process windows.

Despite these challenges, ongoing research in SciML continues to advance the development of more interpretable, computationally efficient, and adaptive modeling strategies that combine physical knowledge with data-driven inference. While significant gaps remain between current capabilities and fully autonomous, closed-loop digital twins, incremental progress in automation, uncertainty awareness, and real-time integration suggests a plausible pathway toward more reliable DT-enabled decision-support systems. In this evolving landscape, physics-informed approaches should currently be regarded as enabling components of future digital twins rather than as standalone solutions, with their long-term potential contingent on sustained advances in model robustness, multiphysics fidelity, and industrial integration.

Author Contributions Fabio Depaoli conducted the majority of the research and was the primary contributor to the writing of the manuscript. Francesco Ponzio and Santa Di Cataldo contributed to the writing and editing of the manuscript. Francesco Ponzio, Enrico Macii, and Santa Di Cataldo critically reviewed the work and provided valuable feedback. All authors read and approved the final manuscript.

Funding Open access funding provided by Politecnico di Torino within the CRUI-CARE Agreement.

Declarations

Conflicts of Interest The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

References

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., & Zheng, X. (2015). TensorFlow: Large-scale machine learning on heterogeneous systems. https://www.tensorflow.org/ (Software available from tensorflow.org).

Aimar, A., Palermo, A., & Innocenti, B. (2019). The role of 3d printing in medical applications: A state of the art. *Journal of Healthcare Engineering, 2019*(1), 5340616. https://doi.org/10.1155/2019/5340616

Anagnostopoulos, S. J., Toscano, J. D., Stergiopulos, N., & Karniadakis, G. E. (2024). Residual-based attention in physics-informed neural networks. *Computer Methods in Applied Mechanics and Engineering, 421*, Article 116805. https://doi.org/10.1016/j.cma.2024.116805

Anagnostopoulos, S. J., Toscano, J. D., Stergiopulos, N., & Karniadakis, G. E. (2026). Learning in pinns: Phase transition, diffusion equilibrium, and generalization. *Neural Networks, 193*, 10798. https://doi.org/10.1016/j.neunet.2025.107983

Anandkumar, A., Azizzadenesheli, K., Bhattacharya, K., Kovachki, N., Li, Z., Liu, B., & Stuart, A. (2019). Neural operator: Graph kernel network for partial differential equations. *Iclr 2020 workshop on integration of deep neural models and differential equations*. https://openreview.net/forum?id=fg2ZFmXFO3.

Annamalai, D., & Nampoothiri, J. (2024). Investigating the effects of ultrasonic assistance on tig welding of aa7075 alloys: a machine

learning-based optimization study using rsm-pso. *Physica Scripta, 100*(1), 01600. https://doi.org/10.1088/1402-4896/ad9554

Annamalai, D., & Nampoothiri, J. (2024). Ultrasonic-assisted tungsten inert gas welding of inconel 625 joints to reduce hot cracking and improve microhardness: Optimization and prediction methods. *Journal of Testing and Evaluation, 52*(4), 2515–2537. https://doi.org/10.1520/JTE20230639

Baidu (2016). Paddlepaddle: An open-source deep learning platform. https://github.com/PaddlePaddle/Paddle. (Online).

Balduzzi, D., Racaniere, S., Martens, J., Foerster, J., Tuyls, K., & Graepel, T. (2018). The mechanics of n-player differentiable games. J. Dy and A. Krause (Eds.), *Proceedings of the 35th international conference on machine learning* (Vol. 80, pp. 354–363). PMLR.

Basri, R., Galun, M., Geifman, A., Jacobs, D., Kasten, Y., & Kritchman, S. (2020). Frequency bias in neural networks for input of non-uniform density. H.D. III and A. Singh (Eds.), *Proceedings of the 37th international conference on machine learning* (Vol. 119, pp. 685–694). PMLR.

Bayat, M., Dong, W., Thorborg, J., To, A. C., & Hattel, J. H. (2021). A review of multi-scale and multi-physics simulations of metal additive manufacturing processes with focus on modeling strategies. *Additive Manufacturing, 47*, 10227. https://doi.org/10.1016/j.addma.2021.102278

Belytschko, T., Liu, W.K., Moran, B., & Elkhodary, K. (2014). Nonlinear finite elements for continua and structures. John wiley & sons.

Bonacina, C., Comini, G., Fasano, A., & Primicerio, M. (1973). Numerical solution of phase-change problems. *International Journal of Heat and Mass Transfer, 16*(10), 1825–183. https://doi.org/10.1016/0017-9310(73)90202-0

Bonev, B., Hundt, C., Kurth, T., Pathak, J., Baust, M., Kashinath, K., & Azizzadenesheli, K. (2023). Modeling earth's atmosphere with spherical fourier neural operators. https://developer.nvidia.com/blog/modeling-earths-atmosphere-with-spherical-fourier-neural-operators/. (Online).

Boyd, J.P. (2001). Chebyshev and fourier spectral methods. Courier Corporation.

Cai, S., Wang, Z., Lu, L., Zaki, T. A., & Karniadakis, G. E. (2021). Deepm&mnet: Inferring the electroconvection multiphysics fields based on operator approximation by neural networks. *Journal of Computational Physics, 436*, Article 110296. https://doi.org/10.1016/j.jcp.2021.110296

Cao, F., Guo, X., Dong, X., & Yuan, D. (2025). wbpinn: Weight balanced physics-informed neural networks for multi-objective learning. *Applied Soft Computing, 170*, Article 112632. https://doi.org/10.1016/j.asoc.2024.112632

Chen, T., & Chen, H. (1995). Universal approximation to nonlinear operators by neural networks with arbitrary activation functions and its application to dynamical systems. *IEEE Transactions on Neural Networks, 6*(4), 911–917. https://doi.org/10.1109/72.392253

Cheng, Z., Li, Z., Xiaoqiang, W., Huang, J., Zhang, Z., Hao, Z., & Su, H. (2025). Accelerating PDE-constrained optimization by the derivative of neural operators. *Forty-second international conference on machine learning*. https://openreview.net/forum?id=LFF7kUQ5Rp.

Cicirello, A. (2024). Physics-enhanced machine learning: a position paper for dynamical systems investigations. *Journal of Physics: Conference Series, 2909*(1), 01203. https://doi.org/10.1088/1742-6596/2909/1/012034

Contributors, P., & NVIDIA. (2025). Nvidia physicsnemo: An open-source framework for physics-based deep learning. https://github.com/NVIDIA/physicsnemo (Online).

Coscia, D., Ivagnes, A., Demo, N., & Rozza, G. (2023). Physics-informed neural networks for advanced modeling. *Journal of Open Source Software*, 8(87), 535. https://doi.org/10.21105/joss.05352.

Cuomo, S., Di Cola, V. S., Giampaolo, F., Rozza, G., Raissi, M., & Piccialli, F. (2022). Scientific machine learning through physics-informed neural networks: Where we are and what's next. *Journal of Scientific Computing, 92*(3), 8. https://doi.org/10.1007/s10915-022-01939-z

Denlinger, E. R., Jagdale, V., Srinivasan, G., El-Wardany, T., & Michaleris, P. (2016). Thermal modeling of inconel 718 processed with powder bed fusion and experimental validation using in situ measurements. *Additive Manufacturing, 11*, 7–1. https://doi.org/10.1016/j.addma.2016.03.003

Depaoli, F., Felicioni, S., Ponzio, F., Aliberti, A., Macii, E., Bondioli, F., & Di Cataldo, S. (2024). Physics-informed neural networks: A step towards data-driven optimization of additive manufacturing. 2024 *IEEE 29th international conference on emerging technologies and factory automation (etfa)* (p. 1–4).

Di Cataldo, S., Vinco, S., Urgese, G., Calignano, F., Ficarra, E., Macii, A., & Macii, E. (2021). Optimizing quality inspection and control in powder bed metal additive manufacturing: Challenges and research directions. *Proceedings of the IEEE, 109*(4), 326–34. https://doi.org/10.1109/JPROC.2021.3054628

Eidell, M., Mukund, R., & Choudhry, S. (2021). Accelerating product development with physics-informed neural networks and nvidia physicsnemo. https://developer.nvidia.com/blog/accelerating-product-development-with-physics-informed-neural-networks-and-modulus/. (Online).

Evans, L.C. (2022). Partial differential equations (Vol. 19). American mathematical society.

Farea, A., Haghani, P., Rossi, F., Molaei, M., El-Khamy, M., & Dessouky, M. (2024). Understanding physics-informed neural networks: Theory, implementation, and applications. AI, 5(3), 74 https://doi.org/10.3390/ai5030074.

Farrag, A., Yang, Y., Cao, N., Won, D., & Jin, Y. (2025). Physics-informed machine learning for metal additive manufacturing. *Progress in Additive Manufacturing, 10*(1), 171–18. https://doi.org/10.1007/s40964-024-00612-1

Frostig, R., Johnson, M.J., & Leary, C. (2019). Compiling machine learning programs via high-level tracing. SysML Conference 2018. Stanford, United States. https://hal.science/hal-05188750.

Gao, W., Xu, R., Deng, Y., & Liu, Y. (2025). Discretization-invariance? on the discretization mismatch errors in neural operators. *The thirteenth international conference on learning representations*. https://openreview.net/forum?id=J9FgrqOOni.

George, R.J., Zhao, J., Kossaifi, J., Li, Z., & Anandkumar, A. (2024). Incremental spatial and spectral learning of neural operators for solving large-scale PDEs. *Transactions on Machine Learning Research*, https://openreview.net/forum?id=xI6cPQObp0.

Gnanasambandam, R., Shen, B., Chung, J., Yue, X., & Kong, Z. (2023). Self-scalable tanh (stan): Multi-scale solutions for physics-informed neural networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 45*(12), 15588–15603. https://doi.org/10.1109/TPAMI.2023.3307688

Goehrke, S. (2018). Additive manufacturing is driving the future of the automotive industry. https://www.forbes.com/sites/sarahgoehrke/2018/12/05/additive-manufacturing-is-driving-the-future-of-the-automotive-industry. (Online).

Greengard, L., & Rokhlin, V. (1997). A new version of the fast multipole method for the laplace equation in three dimensions. *Acta Numerica, 6*, 229–26. https://doi.org/10.1017/S0962492900002725

Grieves, M., & Vickers, J. (2017). Digital twin: Mitigating unpredictable, undesirable emergent behavior in complex systems. In: F.- J. Kahlen, S. Flumerfelt, and A. Alves (Eds.), Transdisciplinary perspectives on complex systems: New findings and approaches (pp. 85–113). Cham: Springer International Publishing.

Hao, Z., Wang, Z., Su, H., Ying, C., Dong, Y., Liu, S., & Zhu, J. (2023). GNOT: A general neural operator transformer for operator learning. A. Krause, E. Brunskill, K. Cho, B. Engelhardt, S. Sabato, and

J. Scarlett (Eds.), Proceedings of the 40th international conference on machine learning (Vol. 202, pp. 12556–12569). PMLR. https://proceedings.mlr.press/v202/hao23c.html.

Hao, Z., Yao, J., Su, C., Su, H., Wang, Z., Lu, F., & Zhu, J. (2024). Pinnacle: A comprehensive benchmark of physics-informed neural networks for solving pdes. A. Globerson et al. (Eds.), Advances in neural information processing systems (Vol. 37, pp. 76721–76774). Curran Associates Inc.

Harandi, A., Moeineddin, A., Kaliske, M., Reese, S., & Rezaei, S. (2024). Mixed formulation of physics-informed neural networks for thermo-mechanically coupled systems and heterogeneous domains. *International Journal for Numerical Methods in Engineering, 125*(4), Article e738. https://doi.org/10.1002/nme.7388

He, J., Koric, S., Kushwaha, S., Park, J., Abueidda, D., & Jasiuk, I. (2023). Novel deeponet architecture to predict stresses in elasto-plastic structures with variable complex geometries and loads. *Computer Methods in Applied Mechanics and Engineering, 415*, Article 116277. https://doi.org/10.1016/j.cma.2023.116277

Hildebrand, S., Nguyen, T., & Eckert, M. (2024). Comparison of neural fem and neural operator methods for applications in solid mechanics: A review. *Neural Computing and Applications, 36*(16), 16657–1668. https://doi.org/10.1007/s00521-024-10132-2

Hu, N., Ding, L., Men, L., Zhou, W., Zhang, W., & Yin, R. (2025). Dual visual inspection for automated quality detection and printing optimization of two-photon polymerization based on deep learning. *Journal of Intelligent Manufacturing, 36*(6), 4025–4037.

Hu, Z., Jagtap, A. D., Karniadakis, G. E., & Kawaguchi, K. (2023). Augmented physics-informed neural networks (apinns): A gating network-based soft domain decomposition methodology. *Engineering Applications of Artificial Intelligence, 126*, 10718. https://doi.org/10.1016/j.engappai.2023.107183

Hughes, T.J. (2003). The finite element method: linear static and dynamic finite element analysis. Courier Corporation.

Jacot, A., Gabriel, F., & Hongler, C. (2018). Neural tangent kernel: Convergence and generalization in neural networks. S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (Eds.), Advances in neural information processing systems (Vol. 31). Curran Associates Inc.

Jagtap, A.D., & Karniadakis, G.E. (2020). Extended physics-informed neural networks (xpinns): A generalized space-time domain decomposition based deep learning framework for nonlinear partial differential equations. Communications in Computational Physics, 28(5), https://doi.org/10.4208/cicp.oa-2020-0164.

Jagtap, A. D., Kawaguchi, K., & Karniadakis, G. E. (2020). Adaptive activation functions accelerate convergence in deep and physics-informed neural networks. *Journal of Computational Physics, 404*, Article 109136. https://doi.org/10.1016/j.jcp.2019.109136

Jagtap, A. D., Kharazmi, E., & Karniadakis, G. E. (2020). Conservative physics-informed neural networks on discrete domains for conservation laws: Applications to forward and inverse problems. *Computer Methods in Applied Mechanics and Engineering, 365*, 11302. https://doi.org/10.1016/j.cma.2020.113028

Javaid, M., & Haleem, A. (2018). Additive manufacturing applications in medical cases: A literature based review. *Alexandria Journal of Medicine, 54*(4), 411–422. https://doi.org/10.1016/j.ajme.2017.09.003

Jia, Y., Naceur, H., Saadlaoui, Y., Dubar, L., & Bergheau, J. (2024). A comprehensive comparison of modeling strategies and simulation techniques applied in powder-based metallic additive manufacturing processes. *Journal of Manufacturing Processes, 110*, 1–2. https://doi.org/10.1016/j.jmapro.2023.12.048

Jin, G., Wong, J. C., Gupta, A., Li, S., & Ong, Y.- S. (2024). Fourier warm start for physics-informed neural networks. *Engineering Applications of Artificial Intelligence,132*, 10788. https://doi.org/10.1016/j.engappai.2024.107887

Jung, J., Kim, H., Shin, H., & Choi, M. (2024). Ceens: Causality-enforced evolutional networks for solving time-dependent partial differential equations. *Computer Methods in Applied Mechanics and Engineering, 427*, 11703. https://doi.org/10.1016/j.cma.2024.117036

Karniadakis, G. E., Kevrekidis, I. G., Lu, L., Perdikaris, P., Wang, S., & Yang, L. (2021). Physics-informed machine learning. *Nature Reviews Physics, 3*(6), 422–440. https://doi.org/10.1038/s42254-021-00314-5

Kazemi Naeini, H., Shomali, R., Pishahang, A., Hasanzadeh, H., Asadi, S., & Gholizadeh Lonbar, A. (2025). Pinn-dt: Optimizing energy consumption in smart building using hybrid physics-informed neural networks and digital twin framework with blockchain security. *Sensors*, 25(19), https://doi.org/10.3390/s25196242.

Kharazmi, E., Zhang, Z., & Karniadakis, G.E. (2019). Variational physics-informed neural networks for solving partial differential equations. https://arxiv.org/abs/1912.00873.

Kissas, G., Yang, Y., Hwuang, E., Witschey, W. R., Detre, J. A., & Perdikaris, P. (2020). Machine learning in cardiovascular flows modeling: Predicting arterial blood pressure from non-invasive 4d flow mri data using physics-informed neural networks. *Computer Methods in Applied Mechanics and Engineering, 358*, Article 112623. https://doi.org/10.1016/j.cma.2019.112623

Kizhakkinan, U., Duong, P. L. T., Laskowski, R., Vastola, G., Rosen, D. W., & Raghavan, N. (2023). Development of a surrogate model for high-fidelity laser powder-bed fusion using tensor train and gaussian process regression. *Journal of Intelligent Manufacturing, 34*(1), 369–385.

Kondor, R., Teneva, N., & Garg, V. (2014). Multiresolution matrix factorization. E.P. Xing and T. Jebara (Eds.), Proceedings of the 31st international conference on machine learning (Vol. 32, pp. 1620–1628). Bejing, China: PMLR.

Kovachki, N., Li, Z., Liu, B., Azizzadenesheli, K., Bhattacharya, K., Stuart, A., & Anandkumar, A. (2023). Neural operator: Learning maps between function spaces with applications to pdes. *Journal of Machine Learning Research*, 24(89), 1–97, http://jmlr.org/papers/v24/21-1524.html.

Kritzinger, W., Karner, M., Traar, G., Henjes, J., & Sihn, W. (2018). Digital twin in manufacturing: A categorical literature review and classification. *IFAC-PapersOnLine,51*(11), 1016–102. https://doi.org/10.1016/j.ifacol.2018.08.474 (16th IFAC Symposium on Information Control Problems in Manufacturing INCOM 2018)

Kushwaha, S., Park, J., Koric, S., He, J., Jasiuk, I., & Abueidda, D. (2024). Advanced deep operator networks to predict multiphysics solution fields in materials processing and additive manufacturing. *Additive Manufacturing, 88*, Article 104266. https://doi.org/10.1016/j.addma.2024.104266

Lagaris, I. E., Likas, A., & Fotiadis, D. I. (1998). Artificial neural networks for solving ordinary and partial differential equations. *IEEE transactions on neural networks, 9*(5), 987–1000. https://doi.org/10.1109/72.712178

LeVeque, R.J. (2002). Finite volume methods for hyperbolic problems (Vol. 31). Cambridge University Press.

Li, S., Wang, T., Sun, Y., & Tang, H. (2025). Multi-physics simulations via coupled fourier neural operator. https://arxiv.org/abs/2501.17296.

Li, Z., Kovachki, N., Azizzadenesheli, K., Liu, B., Stuart, A., Bhattacharya, K., & Anandkumar, A. (2020). Multipole graph neural operator for parametric partial differential equations. H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin (Eds.), Advances in neural information processing systems (Vol. 33, pp. 6755–6766). Curran Associates Inc.

Li, Z., Kovachki, N.B., Azizzadenesheli, K., liu, B., Bhattacharya, K., Stuart, A., & Anandkumar, A. (2021). Fourier neural operator for parametric partial differential equations. *International confer-*

ence on learning representations. https://openreview.net/forum?id=c8P9NQVtmnO.

Li, Z., Zheng, H., Kovachki, N., Jin, D., Chen, H., Liu, B., & Anandkumar, A. (2024). Physics-informed neural operator for learning partial differential equations. *ACM/IMS Journal of Data Science*, 1(3), https://doi.org/10.1145/3648506.

Lin, X., Kunpeng, Z., Fuh, J. Y. H., & Duan, X. (2022). Metal-based additive manufacturing condition monitoring methods: From measurement to control. *ISA Transactions, 120*, 147–166. https://doi.org/10.1016/j.isatra.2021.03.001

Liu, C., Roux, L.L., Ji, Z., Kerfriden, P., Lacan, F., & Bigot, S. (2020). Machine learning-enabled feedback loops for metal powder bed fusion additive manufacturing. *Procedia Computer Science*, 176, 2586–259, https://doi.org/10.1016/j.procs.2020.09.314 (Knowledge-Based and Intelligent Information & Engineering Systems: Proceedings of the 24th International Conference KES2020).

Liu, N., Li, X., Rajanna, M.R., Reutzel, E.W., Sawyer, B., Rao, P., & Yu, Y. (2024). Deep neural operator enabled digital twin modeling for additive manufacturing (Vol. 2) (No. 3).

Liu, Q., Chu, M., & Thuerey, N. (2025). ConFIG: Towards conflict-free training of physics informed neural networks. The thirteenth international conference on learning representations. https://openreview.net/forum?id=APojAzJQiq.

Liu, R., Wang, Z., Sparks, T., Liou, F., & Newkirk, J. (2017). Aerospace applications of laser additive manufacturing. M. Brandt (Ed.), Laser additive manufacturing (p. 351–371). Woodhead Publishing.

Liu, S., Zhang, H., Sun, Y., Wang, X., & Chen, Q. (2025). Architectures, variants, and performance of neural operators: A comprehensive survey. *Neurocomputing, 615*, Article 130518. https://doi.org/10.1016/j.neucom.2025.130518

Lu, L., Jin, P., Pang, G., Zhang, Z., & Karniadakis, G. E. (2021). Learning nonlinear operators via deeponet based on the universal approximation theorem of operators. *Nature machine intelligence, 3*(3), 218–22. https://doi.org/10.1038/s42256-021-00302-5

Lu, L., Meng, X., Mao, Z., & Karniadakis, G. E. (2021). Deepxde: A deep learning library for solving differential equations. *SIAM Review, 63*(1), 208–22. https://doi.org/10.1137/19M1274067

Lu, Y., Liu, C., & Wang, K.I.- K., Huang, H., Xu, X. (2020). Digital twin-driven smart manufacturing: Connotation, reference model, applications and research issues. *Robotics and Computer-Integrated Manufacturing,61*, 10183. https://doi.org/10.1016/j.rcim.2019.101837

Maddu, S., Sturm, D., Müller, C. L., & Sbalzarini, I. F. (2022). Inverse dirichlet weighting enables reliable training of physics informed neural networks. *Machine Learning: Science and Technology, 3*(1), 01502. https://doi.org/10.1088/2632-2153/ac3712

Mankins, J.C. (1995). Technology readiness levels: A white paper (Tech. Rep.). NASA Office of Space Access and Technology.

Mao, Z., Jagtap, A. D., & Karniadakis, G. E. (2020). Physics-informed neural networks for high-speed flows. *Computer Methods in Applied Mechanics and Engineering, 360*, Article 112789. https://doi.org/10.1016/j.cma.2019.112789

Martorelli, M., Gloria, A., Bignardi, C., Calì, M., & Maietta, S. (2020). Design of additively manufactured lattice structures for biomedical applications. *Journal of Healthcare Engineering, 2020*(1), 2707560. https://doi.org/10.1155/2020/2707560

Mathew, T.V., Cherukuri, R., Sethi, H., & Dimitrov, P. (2024). Siemens energy accelerates power grid asset simulation 10,000x using nvidia physicsnemo. https://developer.nvidia.com/blog/spotlight-siemens-energy-accelerates-power-grid-asset-simulation-10000x-using-nvidia-physicsnemo/. (Online).

Mattey, R., & Ghosh, S. (2022). A novel sequential method to train physics informed neural networks for allen cahn and cahn hilliard equations. *Computer Methods in Applied Mechanics and Engineering, 390*, 11447. https://doi.org/10.1016/j.cma.2021.114474

McClenny, L. D., & Braga-Neto, U. M. (2023). Self-adaptive physics-informed neural networks. *Journal of Computational Physics, 474*, Article 111722.

Meng, X., Li, Z., Zhang, D., & Karniadakis, G. E. (2020). Ppinn: Parareal physics-informed neural network for time-dependent pdes. *Computer Methods in Applied Mechanics and Engineering, 370*, 11325. https://doi.org/10.1016/j.cma.2020.113250

Migus, L., Yin, Y., Ahmed Mazari, J., & Gallinari, P. (2022). Multi-scale physical representations for approximating pde solutions with graph neural operators. A. Cloninger et al. (Eds.), Proceedings of topological, algebraic, and geometric learning workshops 2022 (Vol. 196, pp. 332–340). PMLR.

Montalti, A., Ferretti, P., & Santi, G. M. (2024). From cad to g-code: Strategies to minimizing errors in 3d printing process. *CIRP Journal of Manufacturing Science and Technology, 55*, 62–70. https://doi.org/10.1016/j.cirpj.2024.09.005

Moseley, B., Markham, A., & Nissen-Meyer, T. (2023). Finite basis physics-informed neural networks (fbpinns): a scalable domain decomposition approach for solving differential equations. *Advances in Computational Mathematics, 49*(4), 6. https://doi.org/10.1007/s10444-023-10065-9

Mukherjee, T., Wei, H., De, A., & DebRoy, T. (2018). Heat and fluid flow in additive manufacturing—part i: Modeling of powder bed fusion. *Computational Materials Science, 150*, 304–31. https://doi.org/10.1016/j.commatsci.2018.04.022

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., & Chintala, S. (2019). Pytorch: An imperative style, high-performance deep learning library. https://arxiv.org/abs/1912.01703.

Patel, R., Ghosh, T., Wadhawan, A., Li, K., & Anandkumar, A. (2025). A comprehensive comparison of neural operators for 3d industry-scale engineering designs. https://arxiv.org/abs/2510.05995.

Pathak, J., Subramanian, S., Harrington, P., Raja, S., Chattopadhyay, A., Mardani, M., & Anandkumar, A. (2022). Fourcastnet: A global data-driven high-resolution weather model using adaptive fourier neural operators. https://arxiv.org/abs/2202.11214.

Pieressa, A., Baruffa, G., Sorgato, M., & Lucchetta, G. (2025). Enhancing weld line visibility prediction in injection molding using physics-informed neural networks. *Journal of Intelligent Manufacturing, 36*(6), 4305–4318. https://doi.org/10.1007/s10845-024-02460-w

Psichogios, D. C., & Ungar, L. H. (1992). A hybrid neural network-first principles approach to process modeling. *AIChE Journal, 38*(10), 1499–151. https://doi.org/10.1002/aic.690381003

Qi, Q., Tao, F., Hu, T., Anwer, N., Liu, A., Wei, Y., & Nee, A. (2021). Enabling technologies and tools for digital twin. *Journal of Manufacturing Systems, 58*, 3–2. https://doi.org/10.1016/j.jmsy.2019.10.001

Quarteroni, A., & Rozza, G., et al. (2014). Reduced order methods for modeling and computational reduction (Vol. 9). Springer.

Rackauckas, C., Ma, Y., Martensen, J., Warner, C., Zubov, K., Supekar, R., & Edelman, A. (2021). Universal differential equations for scientific machine learning. https://arxiv.org/abs/2001.04385.

Rahaman, N., Baratin, A., Arpit, D., Draxler, F., Lin, M., Hamprecht, F., & Courville, A. (2019a). On the spectral bias of neural networks. International conference on machine learning (pp. 5301–5310).

Rahaman, N., Baratin, A., Arpit, D., Draxler, F., Lin, M., Hamprecht, F., & Courville, A. (2019b). On the spectral bias of neural networks. K. Chaudhuri and R. Salakhutdinov (Eds.), Proceedings of the 36th international conference on machine learning (Vol. 97, pp. 5301–5310). PMLR.

Rahman Chukkan, J., Vasudevan, M., Muthukumaran, S., Ravi Kumar, R., & Chandrasekhar, N. (2015). Simulation of laser butt welding of aisi 316l stainless steel sheet using various heat sources and

experimental validation. *Journal of Materials Processing Technology, 219*, 48–5. https://doi.org/10.1016/j.jmatprotec.2014.12.008

Raissi, M., Perdikaris, P., & Karniadakis, G. E. (2019). 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. https://doi.org/10.1016/j.jcp.2018.10.045

Rohrhofer, F.M., Posch, S., Gößnitzer, C., & Geiger, B.C. (2023). On the role of fixed points of dynamical systems in training physics-informed neural networks. *Transactions on Machine Learning Research*, https://openreview.net/forum?id=56cTmVrg5w.

Ronen, B., Jacobs, D., Kasten, Y., & Kritchman, S. (2019). The convergence rate of neural networks for learned functions of different frequencies. H. Wallach, H. Larochelle, A. Beygelzimer, F. d' Alché-Buc, E. Fox, and R. Garnett (Eds.), Advances in neural information processing systems (Vol. 32). Curran Associates Inc.

Roy, P., & Castonguay, S. T. (2024). Exact enforcement of temporal continuity in sequential physics-informed neural networks. *Computer Methods in Applied Mechanics and Engineering, 430*, 11719. https://doi.org/10.1016/j.cma.2024.117197

Samaniego, E., Anitescu, C., Goswami, S., Nguyen-Thanh, V., Guo, H., Hamdia, K., & Rabczuk, T. (2020). An energy approach to the solution of partial differential equations in computational mechanics via machine learning: Concepts, implementation and applications. *Computer Methods in Applied Mechanics and Engineering, 362*, Article 112790. https://doi.org/10.1016/j.cma.2019.112790

Sharma, R., & Guo, Y. (2025). Thermo-mechanical physics-informed deep learning for prediction of thermal stress evolution in laser metal deposition. *Engineering Applications of Artificial Intelligence, 157*, Article 111554. https://doi.org/10.1016/j.engappai.2025.111554

Shukla, K., Jagtap, A. D., & Karniadakis, G. E. (2021). Parallel physics-informed neural networks via domain decomposition. *Journal of Computational Physics, 447*, Article 110683. https://doi.org/10.1016/j.jcp.2021.110683

Shukla, K., Oommen, V., Peyvan, A., Penwarden, M., Plewacki, N., Bravo, L., & Karniadakis, G. E. (2024). Deep neural operators as accurate surrogates for shape optimization. *Engineering Applications of Artificial Intelligence, 129*, Article 107615. https://doi.org/10.1016/j.engappai.2023.107615

Si, C., & Yan, M. (2025). Convolution-weighting method for the physics-informed neural network: A primal-dual optimization perspective. https://arxiv.org/abs/2506.19805.

Song, Y., Wang, H., Yang, H., Taccari, M. L., & Chen, X. (2024). Loss-attentional physics-informed neural networks. *Journal of Computational Physics, 501*, 11278. https://doi.org/10.1016/j.jcp.2024.112781

Susulovska, N., Nikolaienko, T., Sethi, H., & Nasr, H. (2024). Exploring the potential of physics-informed neural networks for ai applications. https://www.softserveinc.com/en-us/blog/exploring-the-potential-of-physics-informed-neural. (Online).

Tancik, M., Srinivasan, P., Mildenhall, B., Fridovich-Keil, S., Raghavan, N., Singhal, U., & Ng, R. (2020). Fourier features let networks learn high frequency functions in low dimensional domains. H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin (Eds.), Advances in neural information processing systems (Vol. 33, pp. 7537–7547). Curran Associates Inc.

Tao, F., Zhang, H., Liu, A., & Nee, A. Y. C. (2019). Digital twin in industry: State-of-the-art. *IEEE Transactions on Industrial Informatics, 15*(4), 2405–241. https://doi.org/10.1109/TII.2018.2873186

Technology readiness levels handbook for space applications (Tech. Rep.). (2017). European Space Agency.

Tony, A., Badea, I., Yang, C., Liu, Y., Wells, G., Wang, K., & Zhang, W. (2023). The additive manufacturing approach to polydimethylsiloxane (pdms) microfluidic devices: Review and future directions. Polymers, 15(8), https://doi.org/10.3390/polym15081926.

Toscano, J.D., Buendía, M., García, J.C., & del Ser, J. (2024). From PINNs to PIKANs: Recent advances in physics-informed neural networks. https://arxiv.org/abs/2410.13228.

Tuegel, E. J., Ingraffea, A. R., Eason, T. G., & Spottswood, S. M. (2011). Reengineering aircraft structural life prediction using a digital twin. *International Journal of Aerospace Engineering, 2011*(1), Article 154798. https://doi.org/10.1155/2011/154798

Uhlemann, T.H.- J., Lehmann, C., Steinhilper, R. (2017). The digital twin: Realizing the cyber-physical production system for industry 4.0. *Procedia CIRP,61*, 335–340. https://doi.org/10.1016/j.procir.2016.11.152 (The 24th CIRP Conference on Life Cycle Engineering)

Wandke, K., & Zhang, Y. (2024). Drift-correcting multiphysics informed neural network coupled pde solver. *IEEE Journal on Multiscale and Multiphysics Computational Techniques, 9*, 303–311. https://doi.org/10.1109/JMMCT.2024.3452977

Wang, S., & Perdikaris, P. (2023). Long-time integration of parametric evolution equations with physics-informed deeponets. *Journal of Computational Physics, 475*, Article 111855. https://doi.org/10.1016/j.jcp.2022.111855

Wang, S., Sankaran, S., & Perdikaris, P. (2024). Respecting causality for training physics-informed neural networks. *Computer Methods in Applied Mechanics and Engineering, 421*, Article 116813. https://doi.org/10.1016/j.cma.2024.116813

Wang, S., Teng, Y., & Perdikaris, P. (2021). Understanding and mitigating gradient flow pathologies in physics-informed neural networks. *SIAM Journal on Scientific Computing, 43*(5), A3055–A308. https://doi.org/10.1137/20M1318043

Wang, S., Wang, H., & Perdikaris, P. (2021). On the eigenvector bias of fourier feature networks: From regression to solving multi-scale pdes with physics-informed neural networks. *Computer Methods in Applied Mechanics and Engineering, 384*, Article 113938. https://doi.org/10.1016/j.cma.2021.113938

Wang, S., Yu, X., & Perdikaris, P. (2022). When and why pinns fail to train: A neural tangent kernel perspective. *Journal of Computational Physics, 449*, Article 110768. https://doi.org/10.1016/j.jcp.2021.110768

Wang, Y., Peng, Y., Hu, Z., & Li, Y. (2025). Exploring prototype-guided strategy for domain decomposition in physics-informed neural network. *Nonlinear Dynamics, 113*(11), 12487–12514. https://doi.org/10.1007/s11071-025-10871-4

Wang, Y., Yao, Y., Guo, J., & Gao, Z. (2024). A practical pinn framework for multi-scale problems with multi-magnitude loss terms. *Journal of Computational Physics, 510*, Article 113112. https://doi.org/10.1016/j.jcp.2024.113112

Wessels, H., Weißenfels, C., & Wriggers, P. (2020). The neural particle method – an updated lagrangian physics informed neural network for computational fluid dynamics. *Computer Methods in Applied Mechanics and Engineering, 368*, Article 113127. https://doi.org/10.1016/j.cma.2020.113127

Willy, H. J., Li, X., Chen, Z., Herng, T. S., Chang, S., Ong, C. Y. A., & Ding, J. (2018). Model of laser energy absorption adjusted to optical measurements with effective use in finite element simulation of selective laser melting. *Materials & Design, 157*, 24–34. https://doi.org/10.1016/j.matdes.2018.07.029

Yan, C., Gong, H., et al. (2024). Scientific machine learning in metal additive manufacturing: Current progress and future challenges. *Materials & Design, 237*, Article 112455. https://doi.org/10.1016/j.matdes.2023.112455

Yang, S., Peng, S., Guo, J., & Wang, F. (2024). A review on physics-informed machine learning for monitoring metal additive manufacturing process. *Advanced Manufacturing, 1*(2), 1–2. https://doi.org/10.1016/j.jmapro.2024.11.066

Yu, J., Lu, L., Meng, X., & Karniadakis, G. E. (2022). Gradient-enhanced physics-informed neural networks for forward and inverse pde problems. *Computer Methods in Applied Mechanics*

and Engineering, 393, 11482. https://doi.org/10.1016/j.cma.2022.114823

Yuan, B., Wang, H., Song, Y., Heitor, A., & Chen, X. (2025). High-fidelity multiphysics modelling for rapid predictions using physics-informed parallel neural operator. https://arxiv.org/abs/2502.19543.

Zhang, D., Guo, L., & Karniadakis, G. E. (2020). Learning in modal space: Solving time-dependent stochastic pdes using physics-informed neural networks. *SIAM Journal on Scientific Computing, 42*(2), A639–A66. https://doi.org/10.1137/19M1260141

Zhang, G., Yang, H., Zhu, F., & Chen, Y., et al. (2023). Dasa-pinns: Differentiable adversarial self-adaptive pointwise weighting scheme for physics-informed neural networks. https://ssrn.com/abstract=4376049.

Zhou, A., Lorsung, C., Hemmasian, A., & Farimani, A.B. (2024). Strategies for pretraining neural operators. Transactions on Machine Learning Research,, https://openreview.net/forum?id=9vEVeX9oIv.

Zhu, Q., Liu, Z., & Yan, J. (2021). Machine learning for metal additive manufacturing: predicting temperature and melt pool fluid dynamics using physics-informed neural networks. *Computational Mechanics,67*(2), 619–63. https://doi.org/10.1007/s00466-020-01952-9

Zubov, K., McCarthy, Z., Ma, Y., Calisto, F., Pagliarino, V., Azeglio, S., & Rackauckas, C. (2021). Neuralpde: Automating physics-informed neural networks (pinns) with error approximations. arxiv:2107.09443.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.