Physics-guided machine learning is unlocking new capabilities in modeling complex systems

Kishan Prakash[1], Karthika Nasir[1]

[1]Bharti Centre for Communication, Indian Institute of Technology Bombay, India
kishan.prakash@ee.iitb.ac.in, karthika.nasir@ee.iitb.ac.in

Abstract. Physics-Informed Neural Networks (PINNs) have emerged as a powerful paradigm at the intersection of deep learning and scientific computing, offering a novel approach to solving forward and inverse problems governed by partial differential equations (PDEs). By embedding physical laws into the training of neural networks through constrained loss formulations, PINNs unify data-driven learning with mechanistic modeling. This survey provides an in-depth review of the theoretical foundations, algorithmic advances, and practical applications of PINNs across diverse scientific and engineering disciplines. We begin by exploring the motivation and historical context for physics-informed learning, contrasting it with traditional numerical methods and conventional machine learning models. The core methodologies underlying PINNs—including loss construction, network architecture design, optimization techniques, and extensions to stochastic and multi-physics systems—are discussed in detail. A taxonomy of recent variants, such as variational PINNs, probabilistic PINNs, domain-decomposed PINNs, and operator-learning PINNs, is presented to highlight the rapid diversification of the field. Representative applications in fluid dynamics, materials science, geophysics, biomedical engineering, and climate modeling are examined to demonstrate the real-world impact of PINNs. Despite their promise, PINNs face several challenges, including optimization stiffness, high computational cost, generalization limitations, sensitivity to noise, and lack of theoretical guarantees. We provide a critical analysis of these issues and survey emerging solutions. Looking ahead, we identify future research directions such as foundation models for physics, self-supervised learning, symbolic and probabilistic hybrid frameworks, scalable training strategies, and autonomous scientific discovery. By systematically bridging the gap between data and physical laws, PINNs represent a foundational step toward interpretable, generalizable, and trustworthy scientific machine learning. This review aims to serve as a comprehensive reference for researchers and practitioners seeking to understand and advance the frontiers of physics-informed neural computation.

Keywords: Physics-Informed Neural Networks (PINNs); Scientific Machine Learning; Partial Differential Equations (PDEs); Deep Learning; Surrogate Modeling; Operator Learning; Inverse Problems; Uncertainty Quantification; Data-Driven Modeling; Physics-Based AI; Neural Differential Equations; Computational Physics; Hybrid Modeling; Self-Supervised Learning; Scientific Computing.

1 Introduction

In recent years, the confluence of machine learning (ML) and scientific computing has sparked transformative advances across a multitude of disciplines, including computational physics, fluid dynamics, materials science, and biomedical engineering. Among the emerging paradigms at the intersection of data-driven modeling and physical law enforcement, *Physics-Informed Neural Networks* (PINNs) have garnered significant attention due to their unique capability to integrate governing physical equations—typically expressed as partial differential equations (PDEs)—directly into the structure of learning algorithms. PINNs represent a subset of neural network-based models that leverage the expressive power of deep learning while embedding physical priors into the training process, thereby facilitating the solution of forward and inverse problems with increased fidelity, interpretability, and data efficiency. Traditionally, the numerical solution of PDEs has relied on classical methods such as the finite difference method (FDM), finite volume method (FVM), and finite element method (FEM), which are grounded in rigorous mathematical theory and well-established numerical analysis frameworks. While these methods are powerful and extensively validated, they are often computationally intensive, especially for high-dimensional problems, complex geometries, or systems characterized by multi-scale and multi-physics phenomena. Moreover, these methods may suffer from issues such as the curse of dimensionality, difficulty in handling sparse or noisy observational data, and challenges in model calibration and parameter estimation in the presence of uncertainty [1]. In contrast, machine learning, and particularly deep learning, has demonstrated remarkable capabilities in pattern recognition, function approximation, and generalization from data. Neural networks, especially those with deep and wide architectures, have been theoretically shown to serve as universal function approximators. This makes them well-suited for modeling complex nonlinear mappings between inputs and outputs, as well as for capturing high-dimensional data manifolds. However, purely data-driven ML models often lack the inductive biases necessary to enforce physical constraints, which may lead to unphysical predictions, poor extrapolation behavior, and the need for large amounts of training data to ensure generalization. Physics-informed neural networks aim to bridge this gap by embedding physical knowledge, such as conservation laws, boundary conditions, and constitutive relationships, directly into the loss function of a neural network. This is typically accomplished by leveraging automatic differentiation to evaluate the residuals of the governing equations at collocation points sampled from the domain of interest [2]. The neural network is then trained to minimize a composite loss function that includes both data misfit and physics residual terms. The result is a model that not only interpolates observed data but also satisfies the underlying physical laws, even in regions where data may be scarce or entirely absent [3]. Since their inception, PINNs have been applied to a broad spectrum of problems including, but not limited to, fluid flow governed by the Navier–Stokes equations, heat conduction, wave propagation, solid mechanics, electromagnetism, and reactive transport in porous media [4]. These models have also been extended to handle stochastic differential equations, fractional-order PDEs, and integro-differential systems [5]. Moreover, advances in neural network architectures, such as recurrent neural networks (RNNs), convolutional neural networks (CNNs), graph neural networks (GNNs), and transformers, have further enriched the expressive capacity of PINNs, enabling them to tackle increasingly complex and high-dimensional problems [6]. Beyond forward simulations, PINNs are particularly well-suited for inverse problems, where the goal is to infer unknown parameters, source terms, or boundary conditions from indirect or incomplete measurements [7]. In such settings, PINNs offer a natural framework for data assimilation, model calibration, and uncertainty quantification [8]. They are inherently flexible and can be adapted to incorporate Bayesian priors, variational inference schemes, and ensemble methods, thus providing a probabilistic characterization of model predictions. Despite their promise, the application of PINNs and ML-based physics solvers is not without challenges [9]. Issues such as training instability, ill-conditioning, sensitivity to network initialization, and difficulties in learning stiff or highly nonlinear systems remain active areas of research. Efforts to overcome these limitations include the development of adaptive sampling strategies, domain decomposition methods, hybrid physics-ML frameworks, and transfer learning techniques. Additionally, the theoretical foundations of PINNs, including their approximation capabilities, generalization bounds, and convergence behavior, are subjects of ongoing investigation. This review aims to provide a comprehensive overview of the current landscape of physics-informed machine learning, with a primary focus on PINNs and their variants. We categorize and synthesize the literature according to key methodological innovations, application domains, and computational strategies. We also highlight open problems, benchmark studies, and prospective research directions that are poised to shape the next generation of data-driven scientific computing [10]. Our goal is to furnish researchers and practitioners with a rigorous yet accessible survey that not only elucidates the state-of-the-art but also charts a roadmap for future developments at the nexus of physics and machine intelligence [11].

2 Background

The integration of machine learning techniques with physical modeling has necessitated a solid understanding of both the computational methods traditionally used in scientific computing and the foundations of modern deep learning. In this section, we provide an in-depth overview of the mathematical and algorithmic underpinnings that form the basis for physics-informed neural networks (PINNs). This includes a discussion of differential equations in physical modeling, traditional numerical solvers, key components of deep learning, and the evolution of hybrid physics-ML methodologies leading up to the development of PINNs.

2.1 Differential Equations in Scientific Modeling

Most physical processes in nature and engineering can be described by a set of governing equations derived from first principles, such as conservation laws, constitutive relations, and empirical observations [12]. These governing laws are commonly expressed in the form of ordinary differential equations (ODEs) and partial differential equations (PDEs). For example, the conservation of mass, momentum, and energy leads to the Navier–Stokes equations for fluid dynamics, Maxwell's equations for electromagnetism, and the Schrödinger equation for quantum mechanics [13]. Formally, a PDE is defined as:

$$\mathcal{N}[u(\mathbf{x},t);\boldsymbol{\lambda}] = 0, \quad \mathbf{x} \in \Omega,\ t \in [0,T], \tag{1}$$

where $\mathcal{N}$ is a (potentially nonlinear) differential operator, $u(\mathbf{x},t)$ is the unknown solution field, $\boldsymbol{\lambda}$ denotes physical parameters, and Ω is the spatial domain. Solving such equations requires appropriate initial and boundary conditions:

$$u(\mathbf{x},0) = u_0(\mathbf{x}), \quad \mathbf{x} \in \Omega, \tag{2}$$

$$u(\mathbf{x},t) = g(\mathbf{x},t), \quad \mathbf{x} \in \partial\Omega [14]. \tag{3}$$

2.2 Traditional Numerical Methods

To solve PDEs, classical numerical techniques such as finite difference methods (FDM), finite volume methods (FVM), and finite element methods (FEM) are widely used. These methods discretize the domain and approximate derivatives using algebraic representations [15]. While mature and robust, these methods often face challenges including:

- High computational cost for high-dimensional or multi-scale problems.
- Complex mesh generation for irregular domains.
- Inflexibility when data is sparse or partially missing [16].
- Difficulty incorporating noisy or uncertain measurements.

Furthermore, inverse problems — wherein unknown coefficients or source terms are inferred from data — often lead to ill-posed optimization problems that are sensitive to noise and regularization strategies.

2.3 Neural Networks and Deep Learning

Deep learning models, particularly artificial neural networks (ANNs), have become central to modern machine learning due to their ability to learn complex nonlinear mappings. A neural network can be represented as a composition of functions:

$$\hat{u}(\mathbf{x}) = f_{\boldsymbol{\theta}}(\mathbf{x}) = f^{(L)} \circ f^{(L-1)} \circ \cdots \circ f^{(1)}(\mathbf{x}), \tag{4}$$

where each $f^{(l)}$ represents a layer with learnable parameters $\boldsymbol{\theta}^{(l)}$ [17]. Popular activation functions include ReLU, tanh, and sigmoid, and optimization is typically performed using variants of stochastic gradient descent (SGD) or adaptive methods like Adam. The universal approximation theorem guarantees that, under mild assumptions, neural networks can approximate any continuous function to arbitrary precision, given sufficient width and depth. This expressiveness, along with automatic differentiation and GPU acceleration, makes deep networks highly attractive for modeling complex systems.

2.4 From Data-Driven to Physics-Guided Learning

In classical supervised learning, the objective is to minimize a loss function that quantifies the discrepancy between model predictions and observed data:

$$\mathcal{L}_{\text{data}} = \frac{1}{N} \sum_{i=1}^{N} \| f_{\boldsymbol{\theta}}(\mathbf{x}_i) - y_i \|^2, \tag{5}$$

where $\{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$ are the training data. However, in scientific applications, data may be scarce, expensive to obtain, or corrupted by noise. In such settings, incorporating prior physical knowledge into the learning process becomes crucial. Physics-informed learning seeks to constrain the solution space by enforcing known physical laws. This can be achieved through:

- **Hard constraints**, where physical properties are embedded into the architecture or parameterization of the model.
- **Soft constraints**, where physics is incorporated into the loss function as penalty terms [18].
- **Hybrid models**, where data-driven and physics-based components are integrated.

These strategies pave the way for models that are both interpretable and generalizable, with improved sample efficiency and robustness to noise.

2.5 Genesis of Physics-Informed Neural Networks

The formal concept of Physics-Informed Neural Networks was introduced by Raissi, Perdikaris, and Karniadakis in 2019, building on early ideas of using neural networks to solve differential equations. PINNs depart from black-box ML by incorporating the residual of a governing PDE into the loss function:

$$\mathcal{L}_{\text{PINN}} = \mathcal{L}_{\text{data}} + \lambda_{\text{PDE}} \mathcal{L}_{\text{physics}} + \mathcal{L}_{\text{BC/IC}}, \tag{6}$$

where $\mathcal{L}_{\text{physics}}$ measures the violation of the PDE (via automatic differentiation), and $\mathcal{L}_{\text{BC/IC}}$ enforces boundary/initial conditions. The weighting coefficient λ_{PDE} balances the influence of physical constraints versus empirical data. Since their inception, PINNs have rapidly evolved, with extensions such as:

- **Variational PINNs (VPINNs)** using weak-form residuals [19].
- **Extended PINNs (XPINNs)** employing domain decomposition.
- **Stochastic PINNs (SPINNs)** for uncertainty quantification.
- **Operator learning PINNs** via DeepONets or Fourier neural operators [20].

These innovations aim to address limitations such as training instability, poor convergence in stiff regimes, and high computational demands for large-scale problems.

2.6 Scope of This Survey

This review systematically explores the methodologies, applications, and open challenges in the realm of physics-informed neural networks and physics-aware machine learning [21]. We begin by analyzing the core algorithmic components of PINNs and their theoretical justifications. Subsequently, we review prominent application areas, including fluid dynamics, solid mechanics, climate modeling, and biomedical systems. We also discuss recent advances in operator learning, surrogate modeling, and multi-fidelity frameworks. Finally, we identify critical bottlenecks and propose potential research directions for enhancing the scalability, accuracy, and robustness of PINN-based approaches.

3 Methodology of Physics-Informed Neural Networks

Physics-Informed Neural Networks (PINNs) are designed to solve forward and inverse problems governed by differential equations by embedding physical constraints directly into the learning process. In this section, we dissect the mathematical and algorithmic foundations of PINNs, describing how they are constructed, trained, and deployed for scientific inference. We begin with the canonical formulation of PINNs for deterministic PDEs and then extend to advanced variants that address limitations such as training stiffness, generalization to unseen regimes, and computational scalability.

3.1 Canonical PINN Framework

The fundamental idea of PINNs is to approximate the solution $u(\mathbf{x}, t)$ of a PDE using a neural network $u_{\boldsymbol{\theta}}(\mathbf{x}, t)$ with parameters $\boldsymbol{\theta}$. The network is trained to minimize a composite loss function consisting of data-fitting terms and physics-informed residuals. Consider a general PDE of the form:

$$\mathcal{N}[u(\mathbf{x}, t)] = 0, \quad (\mathbf{x}, t) \in \Omega \times [0, T], \tag{7}$$

subject to initial and boundary conditions:

$$u(\mathbf{x}, 0) = u_0(\mathbf{x}), \quad \mathbf{x} \in \Omega, \tag{8}$$

$$u(\mathbf{x}, t) = g(\mathbf{x}, t), \quad \mathbf{x} \in \partial\Omega,\ t \in [0, T]. \tag{9}$$

The corresponding PINN is trained to minimize the following loss:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{data}} + \mathcal{L}_{\text{PDE}} + \mathcal{L}_{\text{BC/IC}}, \tag{10}$$

where:

- $\mathcal{L}_{\text{data}}$ penalizes the difference between predicted and observed data [22].
- $\mathcal{L}_{\text{PDE}}$ penalizes the residual of the PDE at collocation points [23].
- $\mathcal{L}_{\text{BC/IC}}$ enforces boundary and initial conditions.

3.2 Automatic Differentiation and Residual Evaluation

A key feature enabling PINNs is automatic differentiation (AD), which allows efficient and exact computation of derivatives of the neural network output with respect to its input [24]. This makes it possible to compute the PDE residual $\mathcal{N}[u_{\boldsymbol{\theta}}]$ symbolically at runtime. Let $\hat{u} = u_{\boldsymbol{\theta}}(\mathbf{x}, t)$ be the neural network output. Then, for a PDE of the form:

$$\mathcal{N}[u] = \partial_t u + \mathcal{F}(u, \nabla u, \nabla^2 u, \dots), \tag{11}$$

we use AD to evaluate:

$$\hat{f}(\mathbf{x}, t) := \mathcal{N}[u_{\boldsymbol{\theta}}(\mathbf{x}, t)]. \tag{12}$$

The loss term is then:

$$\mathcal{L}_{\text{PDE}} = \frac{1}{N_f} \sum_{i=1}^{N_f} \left| \hat{f}(\mathbf{x}_f^{(i)}, t_f^{(i)}) \right|^2, \tag{13}$$

where $\{(\mathbf{x}_f^{(i)}, t_f^{(i)})\}$ are the collocation points sampled throughout the domain.

3.3 Boundary and Initial Condition Enforcement

The enforcement of initial and boundary conditions is critical for the well-posedness of PDE problems [25]. PINNs typically enforce these through soft penalties in the loss function:

$$\mathcal{L}_{\text{BC/IC}} = \frac{1}{N_b} \sum_{i=1}^{N_b} \left| u_{\boldsymbol{\theta}}(\mathbf{x}_b^{(i)}, t_b^{(i)}) - g(\mathbf{x}_b^{(i)}, t_b^{(i)}) \right|^2 + \frac{1}{N_0} \sum_{i=1}^{N_0} \left| u_{\boldsymbol{\theta}}(\mathbf{x}_0^{(i)}, 0) - u_0(\mathbf{x}_0^{(i)}) \right|^2. \tag{14}$$

Alternatively, hard constraints can be applied by constructing neural network architectures that intrinsically satisfy the constraints (e.g., via modified output layers) [26].

3.4 Inverse Problems and Parameter Identification

One of the compelling strengths of PINNs is their capability to solve inverse problems. These include identifying unknown coefficients, source terms, or boundary values from observed data. This is achieved by treating the unknown parameters $\boldsymbol{\lambda}$ as trainable variables, optimized jointly with the network weights:

$$\mathcal{L}_{\text{total}}(\boldsymbol{\theta}, \boldsymbol{\lambda}) = \mathcal{L}_{\text{data}}(\boldsymbol{\theta}) + \mathcal{L}_{\text{PDE}}(\boldsymbol{\theta}, \boldsymbol{\lambda}) + \mathcal{L}_{\text{BC/IC}}(\boldsymbol{\theta})[27]. \tag{15}$$

The result is a physics-constrained regression problem where both the solution field and physical parameters are inferred simultaneously [28].

3.5 Network Architecture and Sampling Strategies

While fully connected feedforward neural networks (FNNs) are commonly used, recent efforts have explored the integration of more sophisticated architectures:

- **ResNets and DenseNets**: To improve gradient flow and reduce training instability.
- **Fourier feature embeddings**: To capture high-frequency solutions more effectively [29].
- **Graph Neural Networks (GNNs)**: For PDEs on unstructured meshes or manifolds.
- **Transformers and Attention Mechanisms**: For capturing long-range dependencies in space-time.

Sampling strategies play a critical role in the accuracy and efficiency of PINNs. Uniform, Latin hypercube, and adaptive sampling are common. Adaptive methods iteratively refine collocation points based on residual magnitude or gradient information to focus learning where the model underperforms.

3.6 Optimization and Training Dynamics

Training PINNs involves minimizing a composite loss, often requiring careful balancing of its constituent terms [30]. Challenges include:

- **Stiffness of PDEs**: Sharp gradients or rapid temporal dynamics can degrade training stability.
- **Loss imbalance**: Physics loss may dominate or vanish due to scaling disparities.
- **Gradient pathologies**: Vanishing or exploding gradients in deep networks[31].

Proposed remedies include:

- **Curriculum learning**: Progressively increasing complexity of training tasks.
- **Gradient normalization and re-weighting**: Techniques such as NTK-based scaling and Sobolev training.
- **Domain decomposition**: Breaking large domains into smaller, independently trained subdomains (e.g., XPINNs) [32].

3.7 Advanced PINN Variants

To extend the applicability of PINNs to more complex scenarios, several advanced methodologies have been proposed:

- **Variational PINNs (VPINNs)**: Leverage weak forms of PDEs to reduce regularity requirements.
- **XPINNs and Domain-Decomposed PINNs**: Train multiple PINNs over subdomains with interface conditions [33].
- **Bayesian PINNs**: Introduce uncertainty quantification via variational inference or dropout ensembles [34].
- **Multi-fidelity and Transfer Learning**: Use coarse-resolution models to accelerate fine-scale learning.

These developments highlight the adaptability of the PINN framework and its capacity to incorporate ideas from variational methods, statistical inference, and multi-resolution modeling.

3.8 Implementation Tools and Frameworks

A number of open-source tools have facilitated the adoption of PINNs, including:

- **DeepXDE** and **SciANN**: Python libraries built on TensorFlow for defining PINNs with symbolic PDEs.
- **NeuralPDE** (Julia): Integrated with the SciML ecosystem for scientific ML.
- **Modulus** by NVIDIA: An industrial-strength framework supporting multiphysics and complex geometries.

These frameworks streamline development, enable large-scale deployments, and integrate seamlessly with scientific workflows [35].

4 Applications of Physics-Informed Neural Networks

Physics-Informed Neural Networks (PINNs) have found widespread application across diverse domains of science and engineering, driven by their ability to seamlessly integrate domain knowledge with observational data [34]. This section presents a detailed survey of key applications of PINNs, highlighting how their core methodology adapts to domain-specific modeling challenges [36]. We categorize these applications into fundamental fields including fluid dynamics, solid mechanics, heat and mass transfer, bioengineering, electromagnetics, geosciences, and emerging interdisciplinary frontiers.

4.1 Computational Fluid Dynamics

In computational fluid dynamics (CFD), PINNs offer a mesh-free alternative for solving the Navier–Stokes equations, which govern the motion of incompressible and compressible fluids:

$$\nabla \cdot \mathbf{u} = 0, \tag{16}$$

$$\partial_t \mathbf{u} + (\mathbf{u} \cdot \nabla)\mathbf{u} = -\nabla p + \nu \nabla^2 \mathbf{u} + \mathbf{f}[37]. \tag{17}$$

Applications include laminar and turbulent flow modeling, boundary layer analysis, vortex dynamics, and flow control. Notable successes include:

- PINNs solving steady-state and transient flows around bluff bodies (e.g., cylinders, airfoils).
- Data assimilation from PIV (particle image velocimetry) measurements to infer pressure fields.
- Learning Reynolds stress closures in turbulence modeling using hybrid PINN-RANS frameworks [38].

Domain decomposition methods (e.g., XPINNs) are often employed for large Reynolds number flows, while Fourier feature networks improve resolution in boundary layers and high-gradient regions.

4.2 Solid Mechanics and Structural Analysis

Solid mechanics problems governed by elastostatic and elastodynamic equations have been tackled using PINNs to solve both forward and inverse problems. These involve:

$$\nabla \cdot \boldsymbol{\sigma} + \mathbf{b} = \rho \ddot{\mathbf{u}}, \tag{18}$$

where $\boldsymbol{\sigma}$ is the stress tensor, $\mathbf{b}$ is the body force, and $\mathbf{u}$ is the displacement vector. Applications include:

- Identifying spatially varying elastic moduli in heterogeneous materials [39].
- Crack detection and fracture modeling via inverse PINNs.
- Real-time structural health monitoring using embedded PINNs trained on vibration sensor data [40].

PINNs provide a differentiable surrogate that enables sensitivity analysis and uncertainty quantification without reliance on traditional finite element meshes.

4.3 Heat Transfer and Thermo-Mechanical Coupling

Heat conduction, convection, and radiation phenomena are traditionally modeled using the heat equation:

$$\partial_t T = \alpha \nabla^2 T + Q, \tag{19}$$

where T is temperature, α is thermal diffusivity, and Q is a source term [41]. PINNs have been applied to:

- Estimating thermal diffusivity from surface temperature measurements.
- Modeling temperature fields in electronic cooling systems and battery packs [42].
- Coupled thermo-mechanical simulations in additive manufacturing and welding [43].

Inverse heat conduction problems—known to be ill-posed—benefit from the regularization effects of the embedded physics priors.

4.4 Bioengineering and Physiology

The complex, multiscale nature of biological systems has made them fertile ground for physics-informed machine learning. Applications include:

- Inferring blood flow and pressure distributions in arteries using patient-specific data.
- Modeling electrophysiological wave propagation in cardiac tissue using PINNs governed by reaction-diffusion PDEs.
- Tumor growth modeling using coupled advection-reaction equations calibrated to MRI data.

In biomedicine, PINNs facilitate non-invasive estimation of quantities that are difficult or impossible to measure directly, such as stress distributions within soft tissues or conductivity maps in brain tissue.

4.5 Electromagnetics and Photonics

Maxwell's equations govern the behavior of electromagnetic fields and are foundational in optics, microwave engineering, and photonic device design:

$$\nabla \times \mathbf{E} = -\partial_t \mathbf{B}, \tag{20}$$

$$\nabla \times \mathbf{H} = \partial_t \mathbf{D} + \mathbf{J}, \tag{21}$$

$$\nabla \cdot \mathbf{D} = \rho, \tag{22}$$

$$\nabla \cdot \mathbf{B} = 0. \tag{23}$$

PINNs have been successfully applied to:

- Inverse design of photonic devices with target field profiles.
- Subsurface radar imaging and inverse scattering problems [44].
- Time-harmonic electromagnetic field reconstruction in complex media [45].

These applications highlight the ability of PINNs to enforce divergence-free and curl-free constraints directly within their architecture or via soft penalties.

4.6 Geophysics and Earth Science Modeling

Geophysical phenomena such as subsurface flow, seismic wave propagation, and geothermal energy distribution are traditionally modeled using PDEs with complex boundary conditions and multi-physics coupling. PINNs have been employed in:

- Seismic inversion and wavefield extrapolation using elastodynamic PINNs [46].
- Modeling groundwater flow and contaminant transport in porous media.
- CO_2 sequestration and reservoir simulation using data-constrained PINNs.

Uncertainty quantification and surrogate modeling of high-dimensional geostatistical systems are facilitated by combining PINNs with generative models or Gaussian process priors.

4.7 Climate Modeling and Environmental Science

The spatiotemporal dynamics of climate systems—governed by Navier–Stokes equations on a rotating sphere with radiative-convective coupling—pose unique challenges [47]. PINNs are emerging as a tool for:

- Learning reduced-order models of atmospheric or oceanic circulation [48].
- Inferring latent dynamics of carbon fluxes and greenhouse gas distributions [49].
- Assimilating satellite observations into physically consistent climate simulations [50].

Their ability to fuse sparse and heterogeneous data with known conservation laws is especially beneficial for planetary-scale modeling.

4.8 Multiphysics and Coupled Systems

Real-world systems often involve the coupling of multiple physical phenomena—such as thermo-fluid-structure interaction, electro-chemo-mechanical dynamics in batteries, or magnetohydrodynamics [51]. PINNs are increasingly used to model such systems through:

- Multi-network architectures with shared latent representations [52].
- Coupled loss functions that enforce consistency across domains.
- Adaptive sampling to balance competing accuracy demands.

The modular and composable nature of neural networks lends itself to scalable, interpretable, and modular solutions for multiphysics problems.

4.9 Comparison with Traditional Solvers

In many domains, PINNs are not intended to replace established numerical solvers but rather complement them. Advantages include:

- Mesh-free computation and easy adaptation to irregular domains.
- Direct access to derivatives and gradients for sensitivity analysis.
- Natural treatment of inverse problems and data assimilation.

However, challenges remain in terms of training stability, computational cost, and scaling to industrial-grade problems. Hybrid approaches combining PINNs with finite element or finite volume methods are an active area of research.

4.10 Summary of Application Trends

The diversity of applications surveyed above highlights the versatility of PINNs [53]. Broadly, their impact can be categorized into:

- **Forward modeling**: Solving PDEs directly from initial/boundary conditions [24].
- **Inverse problems**: Inferring unknown fields or parameters from sparse observations [54].
- **Surrogate modeling**: Accelerating high-fidelity simulations in optimization loops.
- **Data assimilation**: Reconciling physical models with sensor or experimental data.

In the next section, we critically evaluate the strengths and limitations of PINNs across these applications, provide benchmarking insights, and explore how ongoing methodological advances are addressing existing bottlenecks.

5 Benchmarking and Evaluation of PINNs

Benchmarking the performance of Physics-Informed Neural Networks (PINNs) is critical to understanding their capabilities, limitations, and the contexts in which they provide competitive advantages over traditional numerical methods and alternative machine learning approaches. In this section, we explore the criteria, metrics, and protocols used to evaluate PINNs [55]. We also provide an overview of commonly used benchmark problems, examine comparative studies, and analyze the trade-offs that affect model selection and performance [56].

5.1 Evaluation Metrics

The assessment of PINN performance typically involves a combination of physical, statistical, and computational metrics:

- **Relative L^2 error:**
$$\epsilon_{L^2} = \frac{\|u_{\text{pred}} - u_{\text{true}}\|_{L^2}}{\|u_{\text{true}}\|_{L^2}}, \tag{24}$$
where u_{pred} is the PINN prediction and u_{true} is the ground truth.
- **Physics residual error:** Measures the discrepancy of the predicted solution with respect to the governing PDE at sampled collocation points:
$$\epsilon_{\text{PDE}} = \frac{1}{N_f} \sum_{i=1}^{N_f} \left| \mathcal{N}[u_{\boldsymbol{\theta}}](\mathbf{x}_f^{(i)}, t_f^{(i)}) \right|^2. \tag{25}$$
- **Data mismatch error:** Captures the fidelity of the model in fitting the observational or experimental data.
- **Computational efficiency:** Includes training time, inference time, and memory usage.
- **Scalability and robustness:** Evaluated in terms of performance degradation under increased problem complexity, noise, or dimensionality.
- **Generalization performance:** Assesses how well a PINN trained on limited or localized data extrapolates to unseen regions or longer time horizons.

5.2 Canonical Benchmark Problems

A suite of well-understood PDEs and physical systems has emerged as standard testbeds for evaluating PINNs. These include:

- **1D Burgers' equation:** A nonlinear PDE exhibiting shock formation and viscous dissipation [57]. Used for testing robustness to sharp gradients and nonlinearity [58].
- **Heat and diffusion equations:** Often used in forward and inverse problems to assess temporal accuracy and identifiability of parameters [59].
- **Navier–Stokes equations (2D):** Used for steady and unsteady flow fields, including canonical cases like lid-driven cavity flow, channel flow, and flow past cylinders [60].
- **Wave equations:** Serve to evaluate PINNs under hyperbolic dynamics and oscillatory solutions [61].
- **Allen–Cahn and Cahn–Hilliard equations:** Representative of phase-field dynamics and multiscale pattern formation.
- **Laplace and Poisson equations:** Widely used for elliptic PDEs and inverse coefficient identification problems.

These benchmarks facilitate reproducibility, enable quantitative comparison across methodologies, and reveal systematic weaknesses in standard PINN implementations.

5.3 Comparative Studies with Traditional Solvers

PINNs are often compared with conventional solvers such as finite difference (FD), finite element (FEM), and spectral methods [62]. The key findings across studies include:

- **Accuracy:** For simple geometries and smooth solutions, traditional solvers typically outperform PINNs in raw accuracy. However, PINNs are competitive in inverse and data-limited settings.
- **Flexibility:** PINNs offer significant advantages in handling irregular or moving boundaries, parameter inference, and hybrid data–model integration.
- **Computation:** PINNs incur higher upfront training cost but enable rapid inference post-training [63]. They are attractive in applications requiring many-query evaluations (e.g., optimization and uncertainty quantification).
- **Mesh independence:** As mesh-free solvers, PINNs avoid meshing complexities and adapt naturally to high-dimensional problems.

Hybrid approaches, such as physics-constrained surrogate modeling or PINN-augmented finite element methods, are emerging to leverage the strengths of both paradigms.

5.4 Ablation Studies and Design Sensitivity

The performance of PINNs is influenced by numerous architectural and algorithmic design choices. Ablation studies have identified several key factors:

- **Network depth and width:** Deeper networks may capture more complex dynamics but are harder to train due to vanishing gradients and stiffness [64].
- **Activation functions:** Sinusoidal (Fourier) or adaptive activation functions have shown improved expressivity for PDEs with high-frequency components.
- **Loss balancing strategies:** Dynamic weighting of physics and data losses, gradient normalization, and multi-task learning approaches significantly improve convergence.
- **Sampling schemes:** Adaptive and importance sampling based on residual errors outperform uniform sampling in challenging domains.

Empirical evidence suggests that many PINNs underperform not due to fundamental limitations but due to suboptimal hyperparameter tuning and design configurations [65].

5.5 Uncertainty Quantification Benchmarks

Quantifying uncertainty in PINN predictions is essential for scientific decision-making. Benchmarks for probabilistic PINNs include:

- **Bayesian PINNs (B-PINNs):** Evaluate predictive distributions against known stochastic PDE solutions.
- **Ensemble PINNs:** Compare variance across model outputs and test for calibration against ground truth uncertainties.
- **Physics-guided dropout:** Introduces stochastic regularization while retaining physical constraints [66].

Current benchmarks often use synthetic data generated from stochastic Galerkin methods or Monte Carlo simulations to validate UQ performance.

5.6 Large-Scale and High-Dimensional Benchmarks

While many PINN benchmarks are low-dimensional, ongoing efforts aim to validate scalability to:

- High-dimensional PDEs in finance and quantum mechanics.
- Complex geometries from biomedical imaging or additive manufacturing.
- Multiphysics systems requiring simultaneous solution of coupled equations.

These benchmarks necessitate advanced architectures (e.g., deep operators, graph networks) and parallelized training via GPU clusters or distributed computing frameworks.

5.7 Community Challenges and Open Datasets

To standardize evaluation practices, several community-driven initiatives have emerged:

- **PINNBench**: A curated suite of benchmark problems, metrics, and baseline models for reproducible evaluation.
- **ML4Science competitions**: Public challenges hosted by institutions such as NeurIPS and ICML focusing on PINNs and physics-based learning.
- **Open datasets**: Release of experimental datasets from fluid dynamics, structural testing, and remote sensing to test real-world generalization [67].

These resources facilitate fair comparison, encourage innovation, and support systematic benchmarking of next-generation PINNs.

5.8 Summary and Recommendations

Benchmarking PINNs remains an active and evolving field. While early successes on idealized problems have demonstrated their promise, widespread adoption depends on:

- Developing standardized, reproducible benchmarking protocols [68].
- Creating comprehensive suites of real-world test cases.
- Improving interpretability and quantifiability of errors and uncertainties [69].

A key recommendation is the use of multi-faceted evaluation metrics—combining accuracy, physical consistency, computational cost, and robustness—to obtain a holistic assessment of model performance.

6 Challenges and Open Problems

Despite the growing popularity and demonstrated potential of Physics-Informed Neural Networks (PINNs), several significant challenges hinder their broader adoption and performance across real-world applications. This section explores the key limitations of current approaches, identifies open research questions, and outlines the theoretical and practical barriers that must be addressed to achieve robust, scalable, and generalizable physics-informed machine learning frameworks.

6.1 Optimization Difficulties and Gradient Pathologies

One of the most persistent challenges in training PINNs is the presence of ill-conditioned optimization landscapes, which arise due to:

- **Stiffness in the PDEs**: Highly disparate temporal or spatial scales in the underlying physics (e.g., reaction-diffusion or turbulent systems) lead to stiffness in the loss landscape, resulting in vanishing or exploding gradients.
- **Imbalanced loss terms**: The multi-objective nature of PINNs, combining data loss, boundary loss, and PDE residual loss, introduces scale disparities that hinder effective optimization [70]. Dynamic weighting schemes (e.g., gradient normalization or adaptive loss balancing) have been proposed, but no universal solution exists [71].
- **Spectral bias**: Neural networks tend to learn low-frequency components first (known as the Frequency Principle or F-Principle), leading to slow convergence for problems with sharp features or high-frequency dynamics.

These issues manifest in slow convergence, unstable training dynamics, and poor reproducibility across trials and implementations [72].

6.2 Handling High-Dimensional and Complex Geometries

Scaling PINNs to high-dimensional PDEs remains a critical open problem. Challenges include:

- **Curse of dimensionality**: The number of required collocation points grows exponentially with dimensionality, degrading performance in spatiotemporal or stochastic domains.
- **Complex boundaries**: Many real-world problems involve intricate domain geometries, moving interfaces, or multi-domain coupling. Embedding such constraints into the network architecture or loss function is nontrivial.
- **Inefficient sampling**: Uniform random sampling is often inadequate for resolving fine features or regions of high error, and adaptive methods can be computationally expensive or require prior knowledge.

Recent work in domain decomposition (e.g., XPINNs), coordinate transformations, and graph-based methods provides partial solutions but introduces additional complexity in model design and implementation.

6.3 Interpretability and Reliability of Predictions

Unlike classical solvers grounded in numerical analysis, PINNs are data-driven black-box models. This raises several concerns:

- **Lack of interpretability**: Neural networks do not inherently provide insights into the learned physical mechanisms or causal relationships, making them less trustworthy for critical scientific applications.
- **Prediction failure modes**: PINNs can produce physically plausible outputs that are numerically inaccurate or fail to generalize outside the training distribution, especially in extrapolation regimes.
- **Error certification**: There is no systematic way to bound the error or certify the convergence of PINN solutions, especially for inverse problems with ill-posedness or noisy data [73].

Efforts to address this include physics-aware uncertainty quantification, post-hoc sensitivity analysis, and hybrid interpretable–learning frameworks.

6.4 Data Quality and Noise Sensitivity

Although one of the strengths of PINNs is their ability to operate with limited or sparse data, their performance is still highly dependent on:

- **Noise robustness**: Inverse PINNs are particularly susceptible to overfitting noise in observational data, especially when combined with underconstrained physics priors [7].
- **Heterogeneous data integration**: Combining disparate sources (e.g., sensor data, images, time series) remains a challenge, both in terms of loss formulation and network design.
- **Missing data regions**: Learning in partially observed domains (e.g., shadow zones in tomography) often leads to biased solutions unless augmented with strong priors or regularization.

Probabilistic PINNs, denoising autoencoders, and physics-informed generative models represent promising avenues to improve robustness in noisy and incomplete data regimes.

6.5 Computational Cost and Scalability

Training PINNs, particularly for large-scale and high-fidelity problems, can be computationally prohibitive:

- **Training cost**: Unlike traditional solvers that require only one forward simulation, PINNs require extensive gradient-based optimization and backpropagation through complex PDE operators.
- **Hardware constraints**: High-resolution PINNs often require large memory footprints and GPU/TPU resources, limiting their applicability in resource-constrained environments.

– **Parallelism**: Efficient parallelization of PINN training remains underdeveloped compared to traditional numerical solvers[74].

Efforts such as transfer learning, model pruning, and acceleration using physics-guided pretraining are being explored to mitigate these costs [75].

6.6 Lack of Theoretical Guarantees

Theoretical understanding of PINNs is still in its infancy. Open questions include:

– **Convergence theory**: What conditions guarantee convergence of PINNs to the true PDE solution [76]? What is the role of network architecture, data sampling, and PDE regularity?
– **Approximation bounds**: How accurately can neural networks represent solutions of specific PDE classes, especially when solutions contain discontinuities or singularities?
– **Stability analysis**: How do numerical and optimization instabilities interact in dynamic systems [77]? Can we design PINNs with provable stability under perturbations?

Rigorous theoretical analyses and connections to classical approximation theory and numerical analysis are essential to bridge the gap between practice and theory.

6.7 Multi-Physics and Multi-Scale Challenges

Modeling complex systems involving coupled physical processes at different scales remains difficult:

– **Coupling strategies**: Integrating equations from distinct physical domains (e.g., electro-thermo-mechanical models) into a single coherent framework is challenging due to disparate time scales and spatial resolutions.
– **Scale separation**: PINNs often struggle to represent dynamics with localized phenomena (e.g., shock fronts, phase transitions) alongside global effects [78].
– **Modeling interfaces and boundaries**: Accurately resolving discontinuities or material interfaces across physical domains is still an open challenge [79].

Hierarchical models, operator networks, and multi-resolution architectures offer partial solutions but require further validation and standardization.

6.8 Generalization and Transfer Learning

Generalizing PINNs to unseen domains or varying initial and boundary conditions is essential for deployment in real-world scenarios:

- **Transfer across parameter spaces**: How well do PINNs trained on one instance of a PDE generalize to different parameter regimes or forcing conditions?
- **Domain adaptation**: Can PINNs learn universal representations transferable across geometries or boundary conditions?
- **Pretraining and fine-tuning**: Effective strategies for initializing networks with prior physical knowledge or previously learned dynamics remain underexplored [80].

Meta-learning, few-shot PINNs, and operator learning (e.g., DeepONets, Fourier Neural Operators) represent promising directions for enhancing generalizability [81].

6.9 Summary of Key Bottlenecks

Table 1 summarizes the major open challenges and ongoing efforts to address them.

7 Future Directions and Emerging Trends

As Physics-Informed Neural Networks (PINNs) continue to evolve, a number of research frontiers are emerging that aim to address existing challenges and expand their applicability. These developments span algorithmic innovations, theoretical foundations, novel architectures, integration with domain knowledge, and applications in cutting-edge scientific domains[35]. In this section, we outline promising directions that are shaping the future trajectory of PINNs and physics-informed machine learning.

7.1 Operator Learning and Neural PDE Solvers

One of the most transformative ideas in this domain is the shift from solving individual PDE instances to learning *operators*—mappings between function spaces. Unlike traditional PINNs, operator learning methods aim to generalize across different initial/boundary conditions, geometries, or source terms. Notable approaches include:

- **Deep Operator Networks (DeepONets)**: Learn nonlinear operators by decomposing them into a branch network (encoding input functions) and a trunk network (representing the output domain) [82].
- **Fourier Neural Operators (FNOs)**: Exploit fast Fourier transforms and global convolutions in the spectral domain to learn spatially invariant operators efficiently.
- **Multipole Graph Neural Operators**: Extend operator learning to unstructured meshes and complex topologies using message-passing frameworks.

These models enable zero-shot prediction on new PDE configurations, making them attractive for surrogate modeling, parametric studies, and real-time control.

7.2 Physics-Enhanced Foundation Models

Inspired by the success of large language models (LLMs), the field is moving toward building *foundation models* for scientific machine learning. These are large pretrained models that capture general-purpose physical knowledge and can be fine-tuned for downstream tasks. Characteristics include:

- **Multi-modal integration**: Combining simulation data, empirical measurements, symbolic equations, and textual descriptions.
- **Transfer learning across physics domains**: Enabling knowledge sharing between disciplines such as fluid mechanics, electromagnetics, and materials science.
- **In-context learning**: Using few-shot or zero-shot prompting for solving new PDEs, without retraining.

Recent efforts such as SciFormer, GPT-PDE, and SymFormer are early instances of this paradigm, paving the way for a new generation of generalist scientific models [83].

7.3 Integration with Symbolic and Probabilistic Methods

Combining PINNs with symbolic computation and probabilistic reasoning offers opportunities for enhancing interpretability and robustness:

- **Symbolic regression hybrid PINNs**: Use symbolic neural networks or equation discovery tools (e.g., SINDy, AI Feynman) to extract interpretable physical laws from trained models [84].
- **Bayesian PINNs**: Incorporate uncertainty quantification by placing priors over neural network weights or residual functions, enabling principled posterior inference.
- **Physics-guided probabilistic graphical models**: Combine causal structures and latent variable models with deep PINNs for modeling complex systems with partial observability [85].

These approaches move toward explainable AI frameworks that not only predict but also generate verifiable scientific hypotheses.

7.4 Self-Supervised and Unsupervised Physics Learning

As data labeling and ground truth solutions become increasingly scarce in complex systems, self-supervised and unsupervised approaches are gaining traction:

- **Physics-driven contrastive learning**: Exploit invariances, symmetries, or conservation laws to learn representations without explicit supervision [86].
- **Energy-based models**: Encode physical constraints as energy functions and use unsupervised optimization to find minima consistent with observed data.
- **Consistency learning**: Ensure agreement between forward and inverse models or across multiple physical views (e.g., multi-fidelity data) [87].

These paradigms offer scalable alternatives to supervised PINNs, particularly in high-dimensional or poorly instrumented settings.

7.5 Autonomous Scientific Discovery

The integration of PINNs with automated reasoning, experiment design, and robotic systems opens the door to closed-loop scientific discovery pipelines:

- **Active learning with physics priors**: Guide data acquisition based on epistemic uncertainty and physical constraints to maximize information gain [88].
- **Inverse design and optimization**: Use differentiable PINNs to compute gradients with respect to design variables, enabling efficient optimization of materials, structures, and control strategies [89].
- **Human-in-the-loop physics modeling**: Collaboratively refine PINNs based on expert feedback or symbolic constraints provided during training.

This vision enables systems that can formulate, test, and refine physical models autonomously, accelerating the pace of scientific innovation [90].

7.6 Scalable and Efficient Training Strategies

To scale PINNs to industrial applications and large-scale simulations, future work must improve efficiency:

- **Multigrid and multi-fidelity PINNs**: Incorporate hierarchical resolutions and fidelity levels to reduce training cost while maintaining accuracy [80].
- **Curriculum learning**: Gradually introduce complexity in physics loss terms or geometry to stabilize training and improve generalization [91].
- **Hardware acceleration**: Leverage novel hardware (e.g., FPGAs, neuromorphic chips) and parallel training algorithms to support real-time inference and deployment [92].

Techniques from scientific computing, such as model reduction and domain decomposition, are expected to play a growing role in enhancing scalability.

7.7 Theoretical and Mathematical Advancements

To establish PINNs as reliable scientific tools, continued theoretical progress is essential:

- **Error bounds and convergence rates**: Derive tight a priori and a posteriori error estimates, possibly conditioned on PDE structure or network architecture.
- **Generalization theory for PINNs**: Understand how inductive biases and physics constraints affect generalization, especially in underdetermined or ill-posed problems [93].
- **Function space view of deep networks**: Explore links between neural networks and classical approximation spaces (e.g., Sobolev spaces, reproducing kernel Hilbert spaces).

These developments will provide rigorous foundations for model validation and certification in high-stakes scientific domains [94].

7.8 Ethical, Societal, and Educational Considerations

As PINNs enter broader scientific and engineering workflows, it is important to reflect on their societal impact:

– **Ethical deployment**: Ensure fairness, transparency, and accountability in high-impact applications such as climate modeling, biomedical systems, and infrastructure design [95].
– **Accessible tooling**: Develop user-friendly libraries, documentation, and educational platforms to democratize access to physics-informed learning.
– **Interdisciplinary collaboration**: Foster integration across AI, physics, applied mathematics, and engineering disciplines to promote responsible innovation [96].

Promoting open science and reproducibility will be central to building trust and impact in the broader scientific community.

7.9 Summary

The future of PINNs lies at the intersection of data, physics, and learning. With continued advances in architecture design, optimization, theory, and application, physics-informed machine learning has the potential to redefine how we model, simulate, and understand complex systems [97]. By addressing current limitations and embracing interdisciplinary innovation, the next generation of PINNs will play a transformative role in computational science, engineering, and beyond.

8 Conclusion and Outlook

Physics-Informed Neural Networks (PINNs) represent a compelling paradigm shift in the intersection of machine learning and scientific computing. By embedding physical laws directly into the structure or training objective of neural networks, PINNs offer a framework that is both data-efficient and physically consistent—an appealing alternative to purely data-driven models that often lack interpretability and robustness. Over the past few years, this hybrid approach has witnessed a rapid surge in research activity, fueled by advancements in deep learning, automatic differentiation, and numerical analysis.

This survey has provided an extensive overview of the core principles, algorithmic innovations, representative applications, and emerging trends in the PINNs landscape. We have traced the evolution of the field from classical PDE solvers augmented with neural function approximators to more recent developments in operator learning, surrogate modeling, and physics-informed foundation models. The application domains continue to diversify, spanning fluid dynamics, electromagnetics, biomechanics, geophysics, climate science, and beyond. Furthermore, the integration of PINNs with probabilistic reasoning, symbolic AI, and multi-scale modeling frameworks signals a shift toward increasingly general and robust systems for scientific inference and simulation.

Despite these advances, significant challenges remain. Optimization difficulties, poor generalization in high-dimensional settings, sensitivity to noise, and a lack of theoretical guarantees continue to hinder widespread adoption. Addressing these bottlenecks will require not only algorithmic and architectural improvements but also a deeper theoretical understanding of the interaction between neural network expressivity and the structure of physical laws. The development of rigorous error estimates, adaptive sampling techniques, and uncertainty-aware formulations will be essential for building trustworthy and certifiable models.

Looking forward, the convergence of physics-informed learning with other emerging paradigms—such as foundation models, graph neural operators, meta-learning, and self-supervised learning—presents exciting opportunities. These interdisciplinary intersections may yield novel architectures capable of generalizing across PDE families, learning from unstructured scientific corpora, and supporting real-time inference in complex environments. Furthermore, the continued integration of domain knowledge, whether through symbolic priors, symmetry constraints, or conservation laws, is expected to improve interpretability and accelerate learning in data-scarce regimes.

Another promising direction lies in the automation of scientific discovery. PINNs and related methods have the potential to serve as the computational core of autonomous research systems—capable of designing experiments, inferring hidden dynamics, and proposing new theories through iterative cycles of modeling and validation. Coupled with advances in active learning, reinforcement learning, and human-in-the-loop interfaces, such systems could democratize access to high-fidelity modeling tools and catalyze discovery in underexplored domains.

In summary, Physics-Informed Neural Networks are more than a computational tool—they represent a philosophical shift toward models that learn not just from data, but also from our fundamental understanding of the physical world. As this field matures, its impact will likely extend far beyond traditional simulation tasks, influencing how we conceptualize and conduct scientific inquiry itself. The future of PINNs is rich with potential, and realizing this potential will require a collaborative effort across disciplines, uniting machine learning practitioners, domain scientists, applied mathematicians, and engineers. By continuing to bridge data and theory, PINNs promise to play a central role in the next generation of scientific computing.

References

1. Noel E Sharkey and Amanda JC Sharkey. Adaptive generalisation. *Artificial Intelligence Review*, 7(5):313–328, 1993.
2. Jared Willard, Xiaowei Jia, Shaoming Xu, Michael Steinbach, and Vipin Kumar. Integrating scientific knowledge with machine learning for engineering and environmental systems. *ACM Computing Surveys*, 55(4):1–37, 2022.
3. Gregory Kang Ruey Lau, Apivich Hemachandra, See-Kiong Ng, and Bryan Kian Hsiang Low. Pinnacle: Pinn adaptive collocation and experimental points selection. *arXiv preprint arXiv:2404.07662*, 2024.

4. Xiaonan Lai, Shuo Wang, Zhenggang Guo, Chao Zhang, Wei Sun, and Xueguan Song. Designing a shape–performance integrated digital twin based on multiple models and dynamic data: a boom crane example. *Journal of Mechanical Design*, 143(7):071703, 2021.
5. Ehsan Haghighat and Ruben Juanes. Sciann: A keras/tensorflow wrapper for scientific computations and physics-informed deep learning using artificial neural networks. *Computer Methods in Applied Mechanics and Engineering*, 373:113552, 2021.
6. Mohammadamin Mahmoudabadbozchelou and Safa Jamali. Rheology-informed neural networks (rhinns) for forward and inverse metamodelling of complex fluids. *Scientific reports*, 11(1):1–13, 2021.
7. Yang Liu, Wen Liu, Xunshi Yan, Shuaiqi Guo, and Chen-an Zhang. Adaptive transfer learning for pinn. *Journal of Computational Physics*, 490:112291, 2023.
8. Rafael Bischof and Michael A Kraus. Mixture-of-experts-ensemble meta-learning for physics-informed neural networks. In *Proceedings of 33. Forum Bauinformatik*, 2022.
9. Weiqi Ji, Weilun Qiu, Zhiyu Shi, Shaowu Pan, and Sili Deng. Stiff-pinn: Physics-informed neural network for stiff chemical kinetics. *The Journal of Physical Chemistry A*, 125(36):8098–8106, 2021.
10. Abhishek Gupta and BK Mishra. Neuroevolving monotonic pinns for particle breakage analysis. In *2024 IEEE Conference on Artificial Intelligence (CAI)*, pages 993–996. IEEE, 2024.
11. Zhongkai Hao, Jiachen Yao, Chang Su, Hang Su, Ziao Wang, Fanzhi Lu, Zeyu Xia, Yichi Zhang, Songming Liu, Lu Lu, et al. Pinnacle: A comprehensive benchmark of physics-informed neural networks for solving pdes. *arXiv preprint arXiv:2306.08827*, 2023.
12. Shuning Lin and Yong Chen. Gradient-enhanced physics-informed neural networks based on transfer learning for inverse problems of the variable coefficient differential equations. *Physica D: Nonlinear Phenomena*, 459:134023, 2024.
13. Stevo Bozinovski. Reminder of the first paper on transfer learning in neural networks, 1976. *Informatica (Slovenia)*, 44(3), 2020.
14. Zekun Shi, Zheyuan Hu, Min Lin, and Kenji Kawaguchi. Stochastic taylor derivative estimator: Efficient amortization for arbitrary differential operators. *arXiv preprint arXiv:2412.00088*, 2024.
15. Ting Zhang, Rui Yan, Siqian Zhang, Dingying Yang, and Anhao Chen. Application of fourier feature physics-information neural network in model of pipeline conveying fluid. *Thin-Walled Structures*, 198:111693, 2024.
16. Kejun Tang, Xiaoliang Wan, and Chao Yang. Das-pinns: A deep adaptive sampling method for solving high-dimensional partial differential equations. *Journal of Computational Physics*, 476:111868, 2023.
17. Zongren Zou and George Em Karniadakis. L-hydra: Multi-head physics-informed neural networks. *arXiv preprint arXiv:2301.02152*, 2023.
18. Elliott Skomski, Ján Drgoňa, and Aaron Tuor. Automating discovery of physics-informed neural state space models via learning and evolution. In *Learning for Dynamics and Control*, pages 980–991. PMLR, 2021.
19. Qiming Zhu, Zeliang Liu, and Jinhui Yan. Machine learning for metal additive manufacturing: predicting temperature and melt pool fluid dynamics using physics-informed neural networks. *Computational Mechanics*, 67:619–635, 2021.
20. François Mazé and Faez Ahmed. Diffusion models beat gans on topology optimization. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pages 9108–9116, 2023.

21. Jian Cheng Wong, Pao-Hsiung Chiu, Chinchun Ooi, My Ha Dao, and Yew-Soon Ong. Lsa-pinn: Linear boundary connectivity loss for solving pdes on complex geometry. In *2023 International Joint Conference on Neural Networks (IJCNN)*, pages 1–10. IEEE, 2023.
22. Alex Nichol, Joshua Achiam, and John Schulman. On first-order meta-learning algorithms, 2018.
23. Guo Yuan, Fu Zhuojia, Min Jian, Liu Xiaoting, and Zhao Haitao. Curriculum-transfer-learning based physics-informed neural networks for long-time simulation of nonlinear wave. , 56:1–11, 2023.
24. Xiang Huang, Zhanhong Ye, Hongsheng Liu, Shi Ji, Zidong Wang, Kang Yang, Yang Li, Min Wang, Haotian Chu, Fan Yu, et al. Meta-auto-decoder for solving parametric partial differential equations. *Advances in Neural Information Processing Systems*, 35:23426–23438, 2022.
25. Olga Fuks and Hamdi A Tchelepi. Limitations of physics informed machine learning for nonlinear two-phase transport in porous media. *Journal of Machine Learning for Modeling and Computing*, 1(1), 2020.
26. Andrea Bonfanti, Giuseppe Bruno, and Cristina Cipriani. The challenges of the nonlinear regime for physics-informed neural networks. *arXiv preprint arXiv:2402.03864*, 2024.
27. Geoffrey Négiar, Michael W Mahoney, and Aditi S Krishnapriyan. Learning differentiable solvers for systems with hard constraints. *arXiv preprint arXiv:2207.08675*, 2022.
28. Zhiwei Fang and Justin Zhan. Deep physical informed neural networks for metamaterial design. *IEEE Access*, 8:24506–24513, 2019.
29. Levi McClenny and Ulisses Braga-Neto. Self-adaptive physics-informed neural networks using a soft attention mechanism, 2022.
30. Sepp Hochreiter, A Steven Younger, and Peter R Conwell. Learning to learn using gradient descent. In *Artificial Neural Networks—ICANN 2001: International Conference Vienna, Austria, August 21–25, 2001 Proceedings 11*, pages 87–94. Springer, 2001.
31. Yassine Zniyed, Thanh Phuong Nguyen, et al. Efficient tensor decomposition-based filter pruning. *Neural Networks*, 178:106393, 2024.
32. Jian Zhang, Juan Du, Min Zhang, Ze Chen, Hongwu Zhang, and Chaoqun Nie. Aerodynamic performance improvement of a highly loaded compressor airfoil with coanda jet flap. *Journal of Thermal Science*, 31:151–162, 2022.
33. Yifan Wang and Linlin Zhong. Nas-pinn: neural architecture search-guided physics-informed neural network for solving pdes. *Journal of Computational Physics*, 496:112603, 2024.
34. Apostolos F Psaros, Kenji Kawaguchi, and George Em Karniadakis. Meta-learning pinn loss functions. *Journal of computational physics*, 458:111121, 2022.
35. Abdullah Al Noman, Zinat Tasneem, Md Fahad Sahed, SM Muyeen, Sajal K Das, and Firoz Alam. Towards next generation savonius wind turbine: Artificial intelligence in blade design trends and framework. *Renewable and Sustainable Energy Reviews*, 168:112531, 2022.
36. Wei Ma, Zhaocheng Liu, Zhaxylyk A Kudyshev, Alexandra Boltasseva, Wenshan Cai, and Yongmin Liu. Deep learning for the design of photonic structures. *Nature Photonics*, 15(2):77–90, 2021.
37. Abdul Hannan Mustajab, Hao Lyu, Zarghaam Rizvi, and Frank Wuttke. Physics-informed neural networks for high-frequency and multi-scale problems using transfer learning. *Applied Sciences*, 14(8):3204, 2024.

38. Dongkun Zhang, Lu Lu, Ling Guo, and George Em Karniadakis. Quantifying total uncertainty in physics-informed neural networks for solving forward and inverse stochastic problems. *Journal of Computational Physics*, 397:108850, November 2019.
39. Shaan Desai, Marios Mattheakis, Hayden Joy, Pavlos Protopapas, and Stephen Roberts. One-shot transfer learning of physics-informed neural networks. *arXiv preprint arXiv:2110.11286*, 2021.
40. Makoto Takamoto, Timothy Praditia, Raphael Leiteritz, Daniel MacKinlay, Francesco Alesiani, Dirk Pflüger, and Mathias Niepert. Pdebench: An extensive benchmark for scientific machine learning. *Advances in Neural Information Processing Systems*, 35:1596–1611, 2022.
41. Alexandre M Tartakovsky, C Ortiz Marrero, Paris Perdikaris, Guzel D Tartakovsky, and David Barajas-Solano. Physics-informed deep neural networks for learning parameters and constitutive relationships in subsurface flow problems. *Water Resources Research*, 56(5):e2019WR026731, 2020.
42. Xiaoli Chen, Jinqiao Duan, and George Em Karniadakis. Learning and meta-learning of stochastic advection–diffusion–reaction systems from sparse measurements. *European Journal of Applied Mathematics*, 32(3):397–420, 2021.
43. Ruiyang Zhang, Yang Liu, and Hao Sun. Physics-informed multi-lstm networks for metamodeling of nonlinear structures. *Computer Methods in Applied Mechanics and Engineering*, 369:113226, 2020.
44. Xiaodong Cui, Wei Zhang, Zoltán Tüske, and Michael Picheny. Evolutionary stochastic gradient descent for optimization of deep neural networks. *Advances in neural information processing systems*, 31, 2018.
45. Haixu Wu, Huakun Luo, Yuezhou Ma, Jianmin Wang, and Mingsheng Long. Ropinn: Region optimized physics-informed neural networks. *arXiv preprint arXiv:2405.14369*, 2024.
46. Caio Davi and Ulisses Braga-Neto. Multi-objective pso-pinn. In *1st Workshop on the Synergy of Scientific and Machine Learning Modeling@ ICML2023*, 2023.
47. Carlos Ruiz Herrera, Thomas Grandits, Gernot Plank, Paris Perdikaris, Francisco Sahli Costabal, and Simone Pezzuto. Physics-informed neural networks to learn cardiac fiber orientation from multiple electroanatomical maps. *Engineering with Computers*, 38(5):3957–3973, 2022.
48. Ye Tian, Xingyi Zhang, Chao Wang, and Yaochu Jin. An evolutionary algorithm for large-scale sparse multiobjective optimization problems. *IEEE Transactions on Evolutionary Computation*, 24(2):380–393, 2019.
49. Isaac E Lagaris, Aristidis Likas, and Dimitrios I Fotiadis. Artificial neural networks for solving ordinary and partial differential equations. *IEEE transactions on neural networks*, 9(5):987–1000, 1998.
50. Keith T Butler, Daniel W Davies, Hugh Cartwright, Olexandr Isayev, and Aron Walsh. Machine learning for molecular and materials science. *Nature*, 559(7715):547–555, 2018.
51. Raphaël Pellegrin, Blake Bullwinkel, Marios Mattheakis, and Pavlos Protopapas. Transfer learning with physics-informed neural networks for efficient simulation of branched flows. *arXiv preprint arXiv:2211.00214*, 2022.
52. Yihang Gao, Ka Chun Cheung, and Michael K Ng. Svd-pinns: Transfer learning of physics-informed neural networks via singular value decomposition. In *2022 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 1443–1450. IEEE, 2022.
53. Saviz Mowlavi and Saleh Nabi. Optimal control of pdes using physics-informed neural networks. *Journal of Computational Physics*, 473:111731, 2023.

54. Kenneth O Stanley, Jeff Clune, Joel Lehman, and Risto Miikkulainen. Designing neural networks through neuroevolution. *Nature Machine Intelligence*, 1(1):24–35, 2019.
55. Lorien Y Pratt, Jack Mostow, and Candace A Kamm. Direct transfer of learned information among neural networks. In *Proceedings of the ninth National conference on Artificial intelligence-Volume 2*, pages 584–589, 1991.
56. Jaehyeok Doh, Nandhini Raju, Nagarajan Raghavan, David W Rosen, and Samyeon Kim. Bayesian inference-based decision of fatigue life model for metal additive manufacturing considering effects of build orientation and post-processing. *International Journal of Fatigue*, 155:106535, 2022.
57. Lulu Cao, Zexin Lin, Kay Chen Tan, and Min Jiang. Interpretable solutions for multi-physics pdes using t-nngp. 2025.
58. Yong Wang, Da-Qing Yin, Shengxiang Yang, and Guangyong Sun. Global and local surrogate-assisted differential evolution for expensive constrained optimization problems with inequality constraints. *IEEE transactions on cybernetics*, 49(5):1642–1656, 2018.
59. Yanjie Song, He Wang, He Yang, Maria Luisa Taccari, and Xiaohui Chen. Loss-attentional physics-informed neural networks. *Journal of Computational Physics*, 501:112781, 2024.
60. Yanlai Chen and Shawn Koohy. Gpt-pinn: Generative pre-trained physics-informed neural networks toward non-intrusive meta-learning of parametric pdes. *Finite Elements in Analysis and Design*, 228:104047, 2024.
61. An-Min Zou and Zhun Fan. Fixed-time attitude tracking control for rigid spacecraft without angular velocity measurements. *IEEE Transactions on Industrial Electronics*, 67(8):6795–6805, 2019.
62. Bing-Chuan Wang, Zhen-Dong Ji, Yong Wang, Han-Xiong Li, and Zhongmei Li. A physics-informed composite network for modeling of electrochemical process of large-scale lithium-ion batteries. *IEEE Transactions on Industrial Informatics*, 2024.
63. Maziar Raissi, Paris Perdikaris, and George Em Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707, 2019.
64. John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, et al. Highly accurate protein structure prediction with alphafold. *nature*, 596(7873):583–589, 2021.
65. Chaoli Sun, Yaochu Jin, Ran Cheng, Jinliang Ding, and Jianchao Zeng. Surrogate-assisted cooperative swarm optimization of high-dimensional expensive problems. *IEEE Transactions on Evolutionary Computation*, 21(4):644–660, 2017.
66. Zhenkun Wang, Qingfu Zhang, Yew-Soon Ong, Shunyu Yao, Haitao Liu, and Jianping Luo. Choose appropriate subproblems for collaborative modeling in expensive multiobjective optimization. *IEEE Transactions on Cybernetics*, 53(1):483–496, 2021.
67. Handing Wang and Yaochu Jin. A random forest-assisted evolutionary algorithm for data-driven constrained multiobjective combinatorial optimization of trauma systems. *IEEE transactions on cybernetics*, 50(2):536–549, 2018.
68. Yann Ollivier, Ludovic Arnold, Anne Auger, and Nikolaus Hansen. Information-geometric optimization algorithms: A unifying picture via invariance principles. *Journal of Machine Learning Research*, 18(18):1–65, 2017.

69. S Bozinovski. Teaching space: A representation concept for adaptive pattern classification. *COINS Technical Report No. 81-28*, 1981.
70. Stefano Markidis. The old and the new: Can physics-informed deep-learning replace traditional linear solvers? *Frontiers in big Data*, page 92, 2021.
71. Zongyi Li. Neural operator: Learning maps between function spaces. In *2021 Fall Western Sectional Meeting*. AMS, 2021.
72. Abhishek Gupta and Barada Kanta Mishra. Globally optimized dynamic mode decomposition: A first study in particulate systems modelling. *Theoretical and Applied Mechanics Letters*, page 100563, 2024.
73. Jordon Kho, Winston Koh, Jian Cheng Wong, Pao-Hsiung Chiu, and Chin Chun Ooi. Design of turing systems with physics-informed neural networks. In *2022 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 1180–1186. IEEE, 2022.
74. Yassine Zniyed, Thanh Phuong Nguyen, et al. Enhanced network compression through tensor decompositions and pruning. *IEEE Transactions on Neural Networks and Learning Systems*, 2024.
75. Yujin Tang, Yingtao Tian, and David Ha. Evojax: Hardware-accelerated neuroevolution. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, pages 308–311, 2022.
76. James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, and Qiao Zhang. JAX: composable transformations of Python+NumPy programs, 2018.
77. Yun Li and Hiroshi Kashiwagi. High-order volterra model predictive control and its application to a nonlinear polymerisation process. *International Journal of Automation and Computing*, 2:208–214, 2005.
78. Revanth Mattey and Susanta Ghosh. A novel sequential method to train physics informed neural networks for allen cahn and cahn hilliard equations. *Computer Methods in Applied Mechanics and Engineering*, 390:114474, 2022.
79. Nick Zhang, Abhishek Gupta, Zefeng Chen, and Yew-Soon Ong. Multitask neuroevolution for reinforcement learning with long and short episodes. *IEEE Transactions on Cognitive and Developmental Systems*, 2022.
80. Maryam Toloubidokhti, Yubo Ye, Ryan Missel, Xiajun Jiang, Nilesh Kumar, Ruby Shrestha, and Linwei Wang. Dats: Difficulty-aware task sampler for meta-learning physics-informed neural networks. In *The Twelfth International Conference on Learning Representations*, 2023.
81. Utkan Güngördü and JP Kestner. Robust quantum gates using smooth pulses and physics-informed neural networks. *Physical Review Research*, 4(2):023155, 2022.
82. Hua-Ming Huang, Elena Raponi, Fabian Duddeck, Stefan Menzel, and Mariusz Bujny. Topology optimization of periodic structures for crash and static load cases using the evolutionary level set method. *Optimization and Engineering*, 25(3):1597–1630, 2024.
83. Hubert Baty. A hands-on introduction to physics-informed neural networks for solving partial differential equations with benchmark tests taken from astrophysics and plasma physics. *arXiv preprint arXiv:2403.00599*, 2024.
84. Hyuk Lee and In Seok Kang. Neural algorithm for solving differential equations. *Journal of Computational Physics*, 91(1):110–131, 1990.
85. Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2009.

86. Sifan Wang, Shyam Sankaran, and Paris Perdikaris. Respecting causality for training physics-informed neural networks. *Computer Methods in Applied Mechanics and Engineering*, 421:116813, 2024.
87. Joaquim Peiró and Spencer Sherwin. Finite difference, finite element and finite volume methods for partial differential equations. *Handbook of Materials Modeling: Methods*, pages 2415–2446, 2005.
88. Rongye Shi, Zhaobin Mo, and Xuan Di. Physics-informed deep learning for traffic state estimation: A hybrid paradigm informed by second-order traffic models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 540–547, 2021.
89. Chenxi Wu, Min Zhu, Qinyang Tan, Yadhu Kartha, and Lu Lu. A comprehensive study of non-adaptive and residual-based adaptive sampling for physics-informed neural networks. *Computer Methods in Applied Mechanics and Engineering*, 403:115671, 2023.
90. Rainer Storn and Kenneth Price. Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. *Journal of global optimization*, 11:341–359, 1997.
91. Binghang Lu, Christian Moya, and Guang Lin. Nsga-pinn: a multi-objective optimization method for physics-informed neural network training. *Algorithms*, 16(4):194, 2023.
92. Yuyao Chen, Lu Lu, George Em Karniadakis, and Luca Dal Negro. Physics-informed neural networks for inverse problems in nano-optics and metamaterials. *Optics express*, 28(8):11618–11633, 2020.
93. Ge Jin, Jian Cheng Wong, Abhishek Gupta, Shipeng Li, and Yew-Soon Ong. Fourier warm start for physics-informed neural networks. *Engineering Applications of Artificial Intelligence*, 132:107887, 2024.
94. Mohammad Nabi Omidvar, Xiaodong Li, Yi Mei, and Xin Yao. Cooperative co-evolution with differential grouping for large scale optimization. *IEEE Transactions on evolutionary computation*, 18(3):378–393, 2013.
95. Risto Miikkulainen and Stephanie Forrest. A biological perspective on evolutionary computation. *Nature Machine Intelligence*, 3(1):9–15, 2021.
96. Chuwei Wang, Shanda Li, Di He, and Liwei Wang. Is $\mathcal{L}^2$ physics-informed loss always suitable for training physics-informed neural network? *arXiv preprint arXiv:2206.02016*, 2022.
97. Yang Khoo, Jianfeng Lu, and Lexing Ying. Solving parametric partial differential equations using the neural convolution. *SIAM Journal on Scientific Computing*, 43(3):A1697–A1719, 2021.

Table 1: Summary of key challenges in PINNs and potential research directions.

Challenge	**Description**	**Potential Solutions**
Optimization	Gradient pathologies, loss imbalance	Adaptive weighting, curriculum learning, second-order methods
High-dimensionality	Curse of dimensionality, complex domains	Domain decomposition, latent space modeling
Interpretability	Lack of insight into learned models	Explainable PINNs, sensitivity and saliency analysis
Data quality	Sensitivity to noise and sparsity	Probabilistic modeling, robust training, generative priors
Computational cost	Expensive training, poor scalability	Transfer learning, reduced-order modeling, GPU optimization
Theoretical guarantees	Lack of convergence/stability theory	PDE-specific approximation theory, rigorous bounds
Multiphysics modeling	Coupled, multiscale interactions	Modular architectures, hybrid modeling frameworks
Generalization	Transfer to new problems/domains	Meta-learning, operator-based PINNs