Physics-Informed Neural Networks: A Paradigm for Knowledge-Enhanced Deep Learning in Scientific Computing

Thomas Waweru, 577 Industries Inc.

Abstract

This paper presents a comprehensive review of Physics-Informed Neural Networks (PINNs), a hybrid framework that integrates scientific domain knowledge with deep learning architectures. PINNs incorporate physical laws, typically expressed as partial differential equations (PDEs), directly into the neural network training process. This approach addresses fundamental limitations of purely data-driven models in modeling complex physical systems, particularly in scenarios with data scarcity or when physical plausibility is crucial. We examine the mathematical foundations, implementation strategies, and diverse applications of PINNs across fluid dynamics, structural analysis, and other scientific domains. The paper also discusses current challenges in PINN development, including training difficulties, scalability issues, and theoretical gaps, alongside emerging research directions aimed at overcoming these limitations. By synthesizing recent advances and identifying open problems, this review provides a roadmap for researchers and practitioners seeking to leverage the synergy between physical principles and deep learning for scientific discovery and engineering applications.

1. Introduction

1.1 The Intersection of AI and Physical Sciences

The remarkable success of deep learning in fields like computer vision and natural language processing has sparked interest in applying these techniques to scientific and engineering domains [1,2]. However, modeling complex physical systems presents unique challenges that conventional deep learning approaches struggle to address effectively [3]. Physical phenomena are governed by well-established laws and principles that are often expressed mathematically as differential equations. These equations encapsulate centuries of scientific knowledge and provide strong constraints on possible solutions.

Traditional scientific computing relies on numerical methods to solve these equations, while conventional machine learning approaches attempt to learn patterns directly from data without incorporating domain knowledge. Physics-Informed Neural Networks (PINNs) represent a hybrid paradigm that bridges these approaches by embedding physical laws directly into the neural network architecture [4]. This integration creates a symbiotic relationship where physical principles guide the learning process, and deep learning techniques enhance the solution of complex physical problems.

1.2 Limitations of Purely Data-Driven AI in Physical Systems

Despite their success in many domains, purely data-driven AI approaches face several critical limitations when applied to modeling complex physical systems:

Data Requirements: Deep learning models typically require vast amounts of labeled data to achieve acceptable performance [5]. In specialized domains such as advanced manufacturing or aerospace engineering, high-quality datasets may be scarce, especially for rare events or

extreme conditions [6]. The cost and practical challenges of data acquisition represent significant barriers to the application of conventional deep learning in these fields.
Limited Generalization: Data-driven models often perform poorly when extrapolating beyond their training distribution [7]. Physical systems frequently exhibit complex, non-linear behaviors that may not be fully represented in available training data, leading to unreliable predictions in novel situations. This limitation is particularly problematic in engineering and defense applications where models must perform reliably under varying and potentially extreme conditions [8].
Lack of Interpretability: Conventional deep learning models operate as "black boxes," making it difficult to understand the physical reasoning behind their predictions [9]. This opacity undermines trust in critical applications where safety and reliability are paramount. Additionally, purely data-driven models may produce outputs that, while statistically consistent with training data, violate fundamental physical laws such as conservation principles [10].
Computational Complexity: Many physical systems of interest, such as turbulent fluid flows or multiscale material behaviors, are inherently high-dimensional and exhibit complex dynamics [11]. Purely data-driven approaches often struggle to effectively capture these complexities without incorporating domain knowledge, regardless of data availability [12].

1.3 The Emergence of Physics-Informed Neural Networks

Physics-Informed Neural Networks have emerged as a promising solution to these challenges [4]. PINNs integrate prior physical knowledge, typically in the form of partial differential equations (PDEs), directly into the neural network training process. This approach effectively leverages both the flexibility of deep learning and the structured constraints of physical laws.

The core innovation of PINNs lies in formulating the neural network training as a physics-constrained optimization problem. By including PDE residuals in the loss function, PINNs are trained to satisfy both the available data and the governing physical equations simultaneously [13]. This hybrid approach offers several advantages:

1. **Data Efficiency**: By incorporating physical constraints, PINNs can achieve accurate results with significantly less data than purely data-driven methods [14].
2. **Enhanced Generalization**: The embedded physical laws provide a strong inductive bias that improves model performance on unseen scenarios [15].
3. **Physical Consistency**: Solutions generated by PINNs naturally respect the underlying physical principles, avoiding physically impossible predictions [16].
4. **Interpretability**: The integration of physical laws enhances model transparency and facilitates scientific discovery [17].

This paper provides a comprehensive review of Physics-Informed Neural Networks, examining their mathematical foundations, implementation strategies, applications, current challenges, and future research directions. By synthesizing recent advances in this rapidly evolving field, we aim to provide researchers and practitioners with a clear understanding of the capabilities, limitations, and potential of PINNs for scientific and engineering applications.

2. Mathematical Foundations

2.1 Neural Networks as Universal Function Approximators
Neural networks serve as the computational backbone of PINNs, leveraging their universal approximation capabilities to represent complex functional relationships [18]. In the context of PINNs, a neural network with parameters θ approximates the solution u to a physical problem:

$u(x) \approx u\theta(x)$
where x represents the independent variables (e.g., spatial coordinates and time). The approximation $u\theta(x)$ can be constructed using standard neural network architectures, typically comprising multiple fully connected layers with nonlinear activation functions.

The universal approximation theorem establishes that neural networks with sufficient capacity can approximate any continuous function to arbitrary precision [19]. This theoretical foundation supports the use of neural networks for representing solutions to differential equations, which are often complex, nonlinear functions.

2.2 Governing Equations and Boundary Conditions
Physical systems are typically described by partial differential equations (PDEs) of the form:

$\mathcal{L}u(x) = f(x)$ for $x \in \Omega$
where $\mathcal{L}$ is a differential operator, u(x) is the solution, f(x) is a forcing term, and Ω represents the computational domain. The PDE is accompanied by boundary conditions:

$\mathcal{B}u(x) = g(x)$ for $x \in \partial\Omega$
where $\mathcal{B}$ is a boundary operator, g(x) specifies the boundary values, and $\partial\Omega$ denotes the boundary of the domain.

For initial value problems, an additional condition specifies the state at the initial time t_0:
$u(x, t_0) = h(x)$

These equations mathematically encode the physical laws governing the system's behavior, such as conservation of mass, momentum, or energy.

2.3 PINN Loss Function Formulation
The core innovation of PINNs is the incorporation of physical laws into the neural network training through the loss function [4]. A typical PINN loss function comprises several components:

$L(\theta) = \lambda DE \cdot LDE(\theta) + \lambda BC \cdot LBC(\theta) + \lambda IC \cdot LIC(\theta) + \lambda Data \cdot LData(\theta)$
where:

1. **PDE Residual Loss (LDE)**: Measures how well the neural network satisfies the governing differential equation: $LDE(\theta) = (1/N\_r)\ \Sigma|\mathcal{L}u\theta(x\_i) - f(x\_i)|^2$ where $\{x\_i\}i=1,...N\_r$ are collocation points distributed throughout the domain.

2. **Boundary Condition Loss (LBC)**: Quantifies the error in satisfying boundary conditions: $LBC(\theta) = (1/N_b) \Sigma|\mathcal{B}u\theta(x_i) - g(x_i)|^2$ where $\{x_i\}i=1,...N_b$ are points on the domain boundary.
3. **Initial Condition Loss (LIC)**: For time-dependent problems, measures adherence to initial conditions: $LIC(\theta) = (1/N_0) \Sigma|u\theta(x_i, t_0) - h(x_i)|^2$ where $\{x_i\}i=1,...N_0$ are points at the initial time.
4. **Data Loss (LData)**: Incorporates available observational data: $LData(\theta) = (1/N_d) \Sigma|u\theta(x_i) - u_{data}(x_i)|^2$ where $\{x_i, u_{data}(x_i)\}i=1,...N_d$ represents the observed data points.

The coefficients λDE, λBC, λIC, and λData are weighting parameters that balance the contribution of each term to the total loss.

2.4 Automatic Differentiation

A crucial enabling technology for PINNs is automatic differentiation (AD), which allows for the efficient and accurate computation of derivatives required for evaluating the PDE residuals [20]. Unlike numerical differentiation methods that approximate derivatives using finite differences, AD computes exact derivatives through systematic application of the chain rule.

In PINNs, automatic differentiation is used to calculate the derivatives of the neural network output $u\theta(x)$ with respect to the input variables x. For example, for a second-order PDE, derivatives up to $\partial^2 u\theta/\partial x^2$ can be efficiently computed. These derivatives are then used to evaluate the PDE residual at the collocation points.

Modern deep learning frameworks, such as TensorFlow and PyTorch, provide built-in automatic differentiation capabilities, greatly simplifying the implementation of PINNs. This technology enables the seamless integration of differential equations into the neural network training process without resorting to discretization methods used in traditional numerical solvers.

3. PINN Implementation Strategies

3.1 Network Architecture Design

The choice of neural network architecture significantly impacts PINN performance. While early implementations primarily used fully connected feedforward networks [4], recent research has explored more sophisticated architectures:

Fully Connected Networks: The standard architecture consists of multiple fully connected layers with activation functions such as tanh or swish, which exhibit smoothness properties beneficial for approximating physical solutions [21].

Residual Networks (ResNets): These architectures, incorporating skip connections between layers, have demonstrated improved training stability and performance for complex physical systems [22].

Specialization for Physical Problems: Some approaches use specialized architectures designed to capture specific physical properties. For example, Hamiltonian Neural Networks preserve energy conservation [23], while Lagrangian Neural Networks respect variational principles [24].

The width (number of neurons per layer) and depth (number of layers) of the network must be calibrated according to the complexity of the physical problem. Problems with intricate solution features typically require deeper networks with more parameters.

3.2 Sampling Strategies

The distribution of collocation points for evaluating the PDE residuals significantly affects PINN training efficiency and accuracy:

Uniform Sampling: The simplest approach distributes points uniformly across the domain, but may be inefficient for problems with localized features.

Adaptive Sampling: These methods dynamically adjust the distribution of collocation points based on error estimates or solution complexity [25]. Points are concentrated in regions with high PDE residuals or steep gradients.

Physics-Guided Sampling: Domain knowledge can inform sampling strategies, focusing computational resources on physically significant regions [26].

Recent advances in self-adaptive sampling have demonstrated substantial improvements in training efficiency and solution accuracy, particularly for problems with multiscale features or discontinuities [27].

3.3 Loss Balancing Techniques

Balancing the different components of the PINN loss function presents a significant challenge. Several approaches have been developed to address this issue:

Static Weighting: Assigning fixed weights to different loss terms based on prior knowledge or empirical tuning.

Dynamic Weighting: Adaptively adjusting weights during training based on the relative magnitudes or gradients of the loss components [28].

Learning Rate Annealing: Gradually changing the learning rates for different loss components throughout the training process [29].

Gradient-Based Methods: Techniques like gradient normalization or gradient surgery that directly manipulate the gradients to achieve balanced training [30].

Proper loss balancing is crucial for stable training and accurate solutions, especially for problems with competing objectives or vastly different scales.

3.4 Domain Decomposition Methods

For problems with complex geometries or multiscale features, domain decomposition approaches split the computational domain into smaller, more manageable subdomains:

Overlapping Domains: Adjacent subdomains share overlapping regions, with continuity enforced through additional penalty terms in the loss function [31].

Non-Overlapping Domains: Subdomains connect at interfaces, with continuity and flux conditions imposed as constraints [32].

Domain decomposition enhances scalability, allows for parallel training, and improves performance on problems with localized features or geometrical complexity. These methods also enable the application of PINNs to large-scale problems that would be challenging to tackle with a single network.

4. Applications of PINNs

4.1 Fluid Dynamics

PINNs have demonstrated remarkable capabilities in fluid dynamics applications, offering advantages over traditional computational fluid dynamics (CFD) methods in certain scenarios:
Incompressible Flows: PINNs have been successfully applied to solve the Navier-Stokes equations for laminar flows around obstacles, achieving accurate predictions of velocity and pressure fields [33]. The mesh-free nature of PINNs simplifies the handling of complex geometries that would require sophisticated meshing in traditional CFD.
Turbulence Modeling: Recent advances have extended PINNs to turbulent flow regimes, incorporating Reynolds-Averaged Navier-Stokes (RANS) models or directly resolving the relevant scales [34]. These approaches show promise for applications where traditional turbulence models face limitations.
Flow Field Reconstruction: PINNs excel at reconstructing full flow fields from sparse measurements, leveraging physical constraints to infer unobserved quantities [35]. This capability is valuable for experimental data analysis and flow monitoring in real-world systems.

The mesh-free nature of PINNs offers computational advantages for flows with complex geometries or moving boundaries, while their data assimilation capabilities enable novel applications in flow reconstruction and prediction.

4.2 Structural Analysis

In structural mechanics, PINNs provide new approaches to both forward and inverse problems:
Elasticity and Deformation: PINNs can accurately predict stress and strain distributions in structures under various loading conditions, solving both linear and nonlinear elasticity problems [36]. The continuous nature of neural network representations facilitates the computation of derived quantities like stress concentrations.
Damage Detection: For structural health monitoring applications, PINNs have been employed to detect and characterize damage from limited sensor data [37]. By incorporating the physics of damaged structures, these models can infer damage parameters with minimal instrumentation.
Material Characterization: PINNs provide an effective framework for inverse problems in material modeling, identifying constitutive parameters from experimental measurements [38]. This capability streamlines the development and calibration of material models for complex materials.

The ability of PINNs to handle both forward and inverse problems within a unified framework offers advantages for integrated approaches to structural design, analysis, and monitoring.

4.3 Heat Transfer and Diffusion Processes

Heat transfer and diffusion problems represent another important application area for PINNs:

Conduction and Convection: PINNs have been applied to heat conduction problems with complex geometries and boundary conditions, as well as coupled conduction-convection scenarios [39]. Their mesh-free nature simplifies the handling of multimaterial interfaces and irregular domains.
Phase Change: For problems involving phase transitions, PINNs can track moving phase boundaries without requiring explicit interface tracking methods [40]. This advantage is particularly valuable for solidification and melting processes.
Inverse Heat Transfer: PINNs excel at inverse problems such as source identification and boundary condition estimation from internal temperature measurements [41]. These capabilities support applications in thermal management, process control, and failure analysis.

The continuous, differentiable nature of neural network representations facilitates the computation of heat fluxes and other derived quantities, offering advantages over traditional numerical methods, particularly for inverse problems.

4.4 Multiphysics and Coupled Problems

One of the most promising aspects of PINNs is their natural extension to multiphysics problems involving coupled physical phenomena:
Fluid-Structure Interaction: PINNs can simultaneously solve fluid dynamics and structural mechanics equations with appropriate coupling conditions at interfaces [42]. This unified approach avoids the algorithmic complexity of traditional partitioned methods.
Electromagnetics and Wave Propagation: Applications in electromagnetics and acoustic wave propagation demonstrate the versatility of PINNs for handling wave equations coupled with other physical processes [43].
Reactive Transport: For systems involving fluid flow, chemical reactions, and species transport, PINNs provide a framework for solving the coupled equations with appropriate conservation principles [44].

The unified treatment of multiple physics within a single computational framework represents a significant advantage of PINNs over traditional methods that often require complex coupling algorithms between specialized solvers.

5. Comparative Analysis: PINNs vs. Traditional Methods

5.1 Advantages of PINNs

Mesh-Free Approach: PINNs eliminate the need for computational meshes, simplifying the handling of complex geometries and moving boundaries [45]. This advantage is particularly valuable for problems with evolving domains or multiscale features.
Data Integration: PINNs naturally incorporate both physical models and observational data, enabling seamless data assimilation and hybrid modeling [46]. This integration supports applications where partial observations need to be combined with physical principles.
Inverse Problem Capabilities: The differentiable nature of neural networks facilitates parameter estimation and inverse problem solving within the same framework as forward

problems [47]. This unified approach streamlines workflows in scientific and engineering applications.

Uncertainty Quantification: Recent extensions incorporate probabilistic frameworks to quantify uncertainties in predictions and parameters [48]. These capabilities support robust decision-making in engineering design and analysis.

5.2 Limitations and Challenges

Training Difficulties: PINNs often face challenges in training convergence, particularly for problems with multiple scales or sharp features [49]. The optimization landscape can be complex, with many local minima that complicate the training process.

Computational Cost: For forward problems where traditional numerical methods are well-established, PINNs may require more computational resources during the training phase [50]. This limitation becomes less significant for inverse problems or when considering the entire workflow.

Solution Accuracy: While PINNs can achieve high accuracy for smooth solutions, they may struggle with discontinuities, shocks, or highly oscillatory functions without specialized adaptations [51].

Theoretical Understanding: The theoretical foundations of PINNs, including convergence guarantees and error bounds, remain less developed compared to classical numerical methods [52]. This gap affects confidence in their application to critical systems.

5.3 Computational Performance Comparison

Quantitative comparisons between PINNs and traditional numerical methods reveal domain-specific trade-offs:

Forward Problems: For standard forward problems with simple geometries, traditional methods like finite elements or finite volumes often outperform PINNs in terms of computational efficiency and accuracy [53]. However, PINNs become competitive for problems with complex geometries or multiphysics coupling.

Inverse Problems: PINNs demonstrate superior performance for many inverse problems, requiring fewer function evaluations and providing more robust parameter estimates compared to traditional approaches based on repeated forward solves [54].

Real-Time Applications: Once trained, PINNs offer rapid inference capabilities suitable for real-time applications, outperforming traditional methods that require solving systems of equations at each time step [55].

The choice between PINNs and traditional methods ultimately depends on the specific application, with PINNs offering particular advantages for complex geometries, inverse problems, multiphysics coupling, and data integration scenarios.

6. Current Challenges and Research Directions

6.1 Training Convergence and Stability

Training convergence remains a significant challenge for PINNs, particularly for problems with multiple scales or complex physical behaviors:

Optimization Algorithms: Research on specialized optimization methods for PINNs aims to improve training stability and convergence rates [56]. Techniques like curriculum learning, where simpler problems are solved before tackling more complex ones, show promise.
Loss Function Design: Novel loss formulations, including physics-informed normalization and adaptive weighting schemes, address the balance between different physical constraints [57].
Initialization Strategies: Proper initialization of network weights can significantly impact training success, with physics-guided initialization methods showing particular promise [58].

Recent advances in understanding the training dynamics of PINNs provide insights into failure modes and guide the development of more robust training methodologies.

6.2 Scalability to High-Dimensional Problems
Scaling PINNs to high-dimensional problems presents computational and methodological challenges:
Dimensionality Reduction: Techniques such as proper orthogonal decomposition or autoencoders can reduce the effective dimensionality of the problem space [59].
Physics-Guided Parameterization: Domain knowledge can inform the construction of lower-dimensional parameterizations that capture the essential physics of high-dimensional systems [60].
Tensor Network Approaches: Adapted from quantum physics, tensor networks offer efficient representations of high-dimensional functions with exploitable structure [61].

Progress in addressing the curse of dimensionality will expand the applicability of PINNs to complex systems with many degrees of freedom.

6.3 Theoretical Foundations and Error Analysis
The theoretical understanding of PINNs lags behind their practical applications, creating opportunities for foundational research:
Convergence Analysis: Rigorous mathematical analysis of convergence properties and error bounds for PINNs under different conditions [62].
Approximation Theory: Extending classical results in approximation theory to the context of PINNs, particularly for solutions to differential equations [63].
Stability Analysis: Investigating the numerical stability of PINN solutions, especially for time-dependent problems and systems with multiple time scales [64].

Strengthening the theoretical foundations of PINNs will enhance confidence in their application to critical systems and guide methodological improvements.

6.4 Integration with Physical Simulations
Hybrid approaches combining PINNs with traditional simulation methods represent a promising research direction:
Multifidelity Modeling: Frameworks that integrate high-fidelity physical simulations with PINN-based surrogate models to balance accuracy and computational efficiency [65].

Physics-Enhanced Transfer Learning: Transferring knowledge from physical simulations to PINNs to improve generalization capabilities and reduce training data requirements [66].
Simulation-Assisted Training: Using traditional numerical methods to generate training data or provide initial conditions for PINN training [67].

These hybrid approaches leverage the complementary strengths of PINNs and traditional methods to address complex multiscale, multiphysics problems.

7. Future Outlook

7.1 Emerging Applications
The versatility of PINNs continues to drive their expansion into new application domains:
Biological Systems: Modeling complex biological processes, from intracellular dynamics to organ-level physiology, with coupled reaction-diffusion systems [68].
Climate and Earth System Science: Applications to atmospheric and oceanic flows, with potential for improved parameterizations of subgrid processes in climate models [69].
Advanced Materials Design: Accelerating the discovery and characterization of novel materials through PINN-based multiscale modeling and inverse design [70].

These emerging applications highlight the potential of PINNs to transform scientific computing across diverse domains by integrating physical principles with data-driven approaches.

7.2 Methodological Innovations
Several promising methodological directions are likely to shape the future development of PINNs:
Operator Learning: Extensions beyond function approximation to directly learn operators mapping between function spaces, enabling more efficient solution of families of related PDEs [71].
Symmetry-Preserving Architectures: Network designs that intrinsically respect physical symmetries and conservation laws, improving both accuracy and generalization [72].
Self-Supervised Learning: Leveraging unlabeled data and physical constraints for training PINNs without explicit supervision [73].

These innovations aim to enhance the efficacy, efficiency, and applicability of PINNs across scientific and engineering domains.

7.3 Interdisciplinary Opportunities
The interdisciplinary nature of PINNs creates opportunities for cross-fertilization between fields:
Computational Mathematics and Scientific Computing: Exchange of ideas between traditional numerical methods and neural network approaches, leading to hybrid algorithms with complementary strengths [74].
Statistical Physics and Machine Learning: Insights from statistical physics informing the design and analysis of PINN architectures and training dynamics [75].

Scientific Discovery and Theory Development: PINNs as tools for automated discovery of physical laws and mechanisms from data, potentially accelerating scientific progress [76].
The integration of domain knowledge from physics, mathematics, and computer science will continue to drive innovations in the theory and application of PINNs.

8. Conclusion

Physics-Informed Neural Networks represent a paradigm shift in scientific computing, seamlessly integrating the expressivity of deep learning with the structured constraints of physical laws. By embedding domain knowledge directly into the learning process, PINNs address critical limitations of purely data-driven approaches for modeling complex physical systems, including data scarcity, limited generalization, and physical implausibility.

The mathematical foundations, implementation strategies, and diverse applications reviewed in this paper demonstrate the versatility and potential of PINNs across multiple scientific and engineering domains. While challenges remain in training convergence, scalability, and theoretical understanding, ongoing research continues to enhance the capabilities and reliability of these methods.

Looking forward, the continued development of PINNs and related physics-informed machine learning approaches promises to transform scientific computing by enabling more efficient, accurate, and flexible modeling of complex physical systems. This transformation will support advances in critical applications ranging from engineering design and analysis to scientific discovery and understanding.

By bridging the historical divide between first-principles modeling and data-driven approaches, PINNs exemplify a broader trend toward knowledge-enhanced artificial intelligence that leverages both human understanding of physical principles and the pattern-recognition capabilities of modern machine learning. This synergistic approach holds tremendous potential for accelerating progress in addressing complex scientific and engineering challenges.

References

[1] Goodfellow I, Bengio Y, Courville A. Deep Learning. MIT Press; 2016.
[2] LeCun Y, Bengio Y, Hinton G. Deep learning. Nature. 2015;521(7553):436-444.
[3] Baker N, Alexander F, Bremer T, et al. Workshop report on basic research needs for scientific machine learning: Core technologies for artificial intelligence. U.S. Department of Energy; 2019.
[4] Raissi M, Perdikaris P, Karniadakis GE. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. Journal of Computational Physics. 2019;378:686-707.
[5] Sun L, Han G, Xia Y, et al. Data requirements for machine learning in scientific problems. arXiv preprint arXiv:2102.11309. 2021.
[6] Karpatne A, Atluri G, Faghmous JH, et al. Theory-guided data science: A new paradigm for scientific discovery from data. IEEE Transactions on Knowledge and Data Engineering. 2017;29(10):2318-2331.

[7] Kim JK, Lee K, Lee D, Jin SY, Park N. DPM: A novel training method for physics-informed neural networks in extrapolation. In: Proceedings of the AAAI Conference on Artificial Intelligence. 2021;35(13):11684-11692.
[8] Stewart R, Ermon S. Label-free supervision of neural networks with physics and domain knowledge. In: Proceedings of the AAAI Conference on Artificial Intelligence. 2017;31(1).
[9] Rudin C. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. Nature Machine Intelligence. 2019;1(5):206-215.
[10] Yang Y, Perdikaris P. Adversarial uncertainty quantification in physics-informed neural networks. Journal of Computational Physics. 2019;394:136-152.
[11] Duraisamy K, Iaccarino G, Xiao H. Turbulence modeling in the age of data. Annual Review of Fluid Mechanics. 2019;51:357-377.
[12] Wang R, Kashinath K, Mustafa M, Albert A, Yu R. Towards physics-informed deep learning for turbulent flow prediction. In: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. 2020:1457-1466.
[13] Karniadakis GE, Kevrekidis IG, Lu L, Perdikaris P, Wang S, Yang L. Physics-informed machine learning. Nature Reviews Physics. 2021;3(6):422-440.
[14] Raissi M. Deep hidden physics models: Deep learning of nonlinear partial differential equations. Journal of Machine Learning Research. 2018;19(1):932-955.
[15] Jagtap SS, Kharazmi E, Karniadakis GE. Conservative physics-informed neural networks on discrete domains for conservation laws: Applications to forward and inverse problems. Computer Methods in Applied Mechanics and Engineering. 2020;365:113028.
[16] Pang G, Lu L, Karniadakis GE. fPINNs: Fractional physics-informed neural networks. SIAM Journal on Scientific Computing. 2019;41(4)
.
[17] Willard J, Jia X, Xu S, Steinbach M, Kumar V. Integrating physics-based modeling with machine learning: A survey. ACM Computing Surveys. 2020;1(1):1-34.
[18] Hornik K, Stinchcombe M, White H. Multilayer feedforward networks are universal approximators. Neural Networks. 1989;2(5):359-366.
[19] Cybenko G. Approximation by superpositions of a sigmoidal function. Mathematics of Control, Signals, and Systems. 1989;2(4):303-314.
[20] Baydin AG, Pearlmutter BA, Radul AA, Siskind JM. Automatic differentiation in machine learning: a survey. Journal of Machine Learning Research. 2018;18:1-43.
[21] Lu L, Meng X, Mao Z, Karniadakis GE. DeepXDE: A deep learning library for solving differential equations. SIAM Review. 2021;63(1):208-228.
[22] He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016:770-778.
[23] Greydanus S, Dzamba M, Yosinski J. Hamiltonian neural networks. Advances in Neural Information Processing Systems. 2019;32:15379-15389.
[24] Cranmer M, Greydanus S, Hoyer S, et al. Lagrangian neural networks. arXiv preprint arXiv:2003.04630. 2020.
[25] Nabian MA, Meidani H. A deep neural network surrogate for high-dimensional random partial differential equations. Journal of Computational Physics. 2019;397:108854.
[26] Wu JL, Xiao H, Paterson E. Physics-informed machine learning approach for augmenting turbulence models: A comprehensive framework. Physical Review Fluids. 2018;3(7):074602.

[27] Wight CL, Zhao J. Solving Allen-Cahn and Cahn-Hilliard equations using the adaptive physics informed neural networks. Communications in Computational Physics. 2021;29(1):123-155.
[28] Wang S, Wang H, Perdikaris P. 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. 2021;384:113938.
[29] Wang S, Teng Y, Perdikaris P. Understanding and mitigating gradient flow pathologies in physics-informed neural networks. SIAM Journal on Scientific Computing. 2021;43(5).
[30] Yu J, Lu L, Meng X, Karniadakis GE. Gradient-enhanced physics-informed neural networks for forward and inverse PDE problems. Computer Methods in Applied Mechanics and Engineering. 2022;393:114823.
[31] Jagtap AD, Karniadakis GE. 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. 2020;28(5):2002-2041.
[32] Moseley B, Markham A, Nissen-Meyer T. Finite basis physics-informed neural networks (FBPINNs): a scalable domain decomposition approach for solving differential equations. arXiv preprint arXiv:2107.07871. 2021.
[33] Raissi M, Yazdani A, Karniadakis GE. Hidden fluid mechanics: Learning velocity and pressure fields from flow visualizations. Science. 2020;367(6481):1026-1030.
[34] Mao Z, Jagtap AD, Karniadakis GE. Physics-informed neural networks for high-speed flows. Computer Methods in Applied Mechanics and Engineering. 2020;360:112789.
[35] Cai S, Wang Z, Wang S, Perdikaris P, Karniadakis GE. Physics-informed neural networks for heat transfer problems. Journal of Heat Transfer. 2021;143