

A Physics-Informed Neural Network Approach to Numerical Solution of Partial Differential Equations

Author Name & Affiliations

Anmol Arshad

Department of Mathematics,

Shaheed Benazir Bhutto

Women University, Peshawar

anmolarshad94@gmail.com

Article Processing History:

Received: 12 May 2026

Revised: 30 June 2026

Published: 22 July 2026

Index Terms :

physics-informed neural networks; partial differential equations; scientific machine learning; automatic differentiation; inverse problems; deep learning.

Abstract

Physics-informed neural networks (PINNs) have become a revolutionary computational paradigm that combines the power of data-driven learning with the laws of physics to solve forward and inverse problems of partial differential equations (PDEs). Formally introduced by Raissi et al., PINNs substitute the residue of a PDE, along with the initial and boundary conditions, into the loss function of a neural network, which computes the necessary derivatives through automatic differentiation. Unlike classical numerical solvers, like the finite element method or finite difference method, PINNs are mesh-less, easy to adapt to irregular geometries, can treat high dimensional problems naturally, and provide a natural framework to integrate sparse observational information with the underlying physical model. This review summarizes the mathematical background of PINNs, architectures and training approaches to address their limitations, and the key variants, such as extended PINNs (XPINNs), conservative PINNs (cPINNs), and operator-learning frameworks like DeepONet (Deep Learning Operator Network), and Fourier neural operators. A survey of applications in fluid mechanics, heat transfer, solid mechanics, biomedical modeling and geophysics is conducted. The continued difficulties with training stability, spectral bias, multi-scale problem and theoretical convergence guarantees are discussed and future prospects for adaptive sampling, causal training and hybrid solver–network schemes are highlighted.

INTRODUCTION

Partial differential equations (PDEs) are the mathematical language of the physical sciences, which are used to model phenomena ranging from fluid flow to heat conduction to wave propagation to elasticity to quantum mechanics. Numerous PDEs of practical importance do not have an explicit solution, and numerical techniques that have been developed over the last few decades have been used to find approximate solutions on the equivalent discretized domain: finite differences (FD), finite elements (FEM), finite volumes, and spectral methods (LeVeque, 2007). Although these are classical methods, with well understood and analyzed solutions, they are increasingly challenged with high dimensional problems, problems with complex geometries, or noisy experimental data.

The emergence of scientific machine learning has provided an attractive alternative. Neural networks are universal function approximators and with proper training can represent the solutions of a PDE without using a mesh. Physics-informed neural networks (PINNs) (Raissi, Perdikaris, and Karniadakis, 2019) impose a PDE constraint on the neural network by adding the penalty on the residual at a specified set of collocation points to the initial and boundary conditions. In this way, PINNs are automatically mesh-free and differentiable end-to-end, as we compute the required derivatives of the output of a network exactly, thanks to automatic differentiation (Baydin et al., 2018).

PINNs have been used for solving problems such as Navier–Stokes flows (Jin et al., 2021), hemodynamics (Kissas et al., 2020), seismic inversion (Rasht-Behesht et al., 2022), and many others since the seminal paper. The number of publications on PINNs has skyrocketed, and comprehensive reviews list dozens of variants and thousands of applications (Karniadakis et al., 2021; Cuomo et al., 2022). But there are basic issues of convergence, optimization and scalability to consider. The aim of this paper is to give a self-contained introduction and critical analysis of the current state of the art of PINNs, covering the mathematical framework, important architectural advances, applications, and challenges.

MATHEMATICAL FRAMEWORK OF PINNS

PROBLEM STATEMENT

Consider a general PDE of the form $N[u](x, t) = 0$ defined on a domain $\Omega \subset \mathbb{R}^d$ for $t \in [0, T]$, with initial condition $u(x, 0) = u_0(x)$ and boundary condition $B[u](x, t) = 0$ on $\partial\Omega \times [0, T]$. In this case N and B are (possibly nonlinear) differential operators and $u(x, t)$ is the solution field to be determined. Classical solvers approximate the solution of u on a mesh, whereas PINNs approximate the solution of u by a neural network $\hat{u}_\theta(x, t)$ defined via weights and biases θ (Raissi et al., 2019).

COMPOSITE LOSS FUNCTION

The network is trained by minimizing a weighted sum of loss terms that enforce the PDE, initial condition, boundary condition and (optionally) any available data:

$$L(\theta) = w_{\text{PDE}} \cdot L_{\text{PDE}} + w_{\text{BC}} \cdot L_{\text{BC}} + w_{\text{IC}} \cdot L_{\text{IC}} + w_{\text{data}} \cdot L_{\text{data}}.$$

Each term is the average of the squared residuals for a specific set of sampled points. L_{PDE} , for instance, is defined as the sum of $|N[\hat{u}_\theta](x_i, t_i)|^2$ over N_r collocation points within the domain $(1/N_r) \sum |N[\hat{u}_\theta](x_i, t_i)|^2$. All spatial and temporal

derivatives that enter N are computed automatically, to machine precision, without truncation error since $\hat{u}_\theta$ is a differentiable neural network (Baydin et al., 2018). The general architecture of PINN is shown in Figure 1.

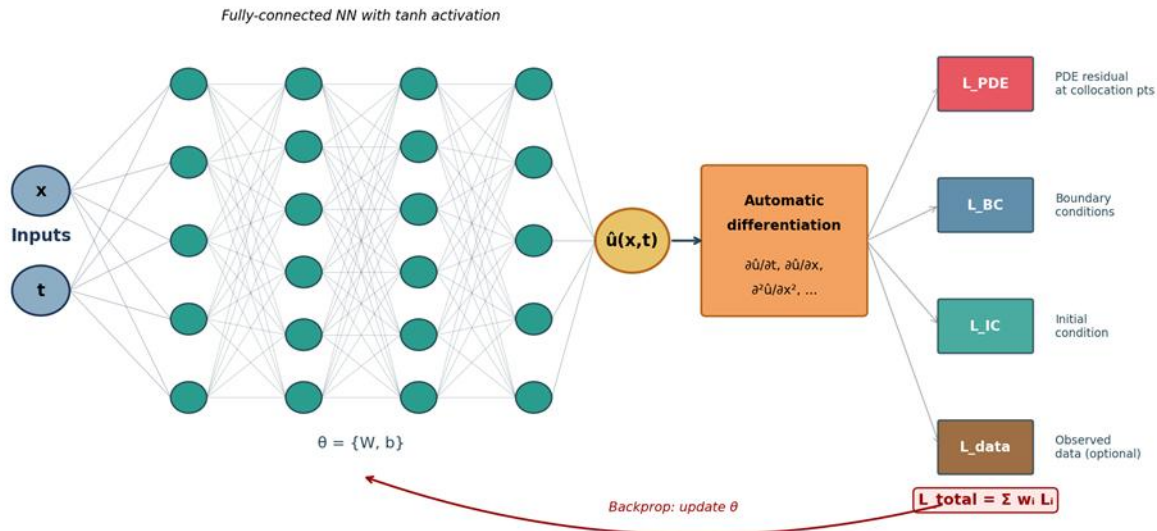


Figure 1: Schematic of a physics-informed neural network (PINN). Spatial and temporal coordinates enter a fully-connected feed-forward network with smooth activations (e.g., tanh). The predicted field $\hat{u}(x, t)$ is passed through automatic differentiation to obtain the derivatives required by the PDE. The total loss combines the PDE residual, boundary and initial conditions, and (optionally) observed data. Backpropagation updates the network parameters θ .

Automatic differentiation

PINNs are technically enabled by automatic differentiation (AD). AD is different from symbolic differentiation: It numerically computes derivatives of a network by systematically using the chain rule through the computational graph of the network; different from numerical differentiation it is exact up to floating point precision (Baydin et al., 2018). The training of deep PINNs is computationally tractable because of reverse-mode AD, which is implemented in frameworks like TensorFlow, PyTorch and JAX, and which performs a single backward pass to compute the gradient of a scalar loss with respect to millions of network parameters (Chaudhari, 2026).

Forward vs. inverse problems

In forward problems, the PDE coefficients and boundary conditions are completely known and PINNs are employed to compute the solution field. Inverse problems, such as determining a diffusivity, a wave velocity, or a stress field, involve some coefficients which are unknown and are determined from a few measurements. PINNs are designed to deal with the inverse problem naturally by considering the unknown coefficients as another trainable parameter that is optimized in combination with the weights of the network (Raissi et al., 2019; Chen et al., 2020). The ability to treat forward and inverse problems in a unified manner is one of the key strengths of the PINN framework.

ARCHITECTURES AND TRAINING STRATEGIES

NETWORK ARCHITECTURES

The original paper of Raissi et al. (2019) employed a fully-connected multilayer perceptron with tanh activations, usually 4-8 hidden layers of 20-100 neurons per layer. Smooth activations are preferred as they have higher order derivatives which would not be present under ReLU that are used in the loss. More recent architectures are residual networks, adaptive-activation networks(AAN) (Jagtap et al., 2020a), sinusoidal representation networks (SIREN) which reduce the spectral bias of MLPs, and Fourier feature embeddings (Wang, Wang, & Perdikaris, 2021) which introduce high-frequency information at the input layer.

OPTIMIZATION

The training of a PINN is a non-convex optimization problem, which is ill-conditioned and suffers from imbalance in the loss terms. The typical recipe is to use a first-order optimizer, like Adam, during the initial training process and a quasi-Newton method, like L-BFGS, during the refinement process (Raissi et al., 2019). Recent analyses have revealed the presence of conflicting gradients in the PINN loss landscape between the PDE residual and the boundary/initial constraints, and adaptive weighting schemes like gradient-based normalization (Wang, Teng, & Perdikaris, 2021) or neural tangent kernel weighting (Wang et al., 2022) have proven to be helpful for convergence.

SAMPLING OF COLLOCATION POINTS

The accuracy of PINN is greatly influenced by selection of collocation points. Typical sampling schemes include uniform or Latin-hypercube sampling, but more accurate solutions can be obtained using adaptive methods that focus sampling points in areas where the residual is large, such as residual-based adaptive refinement (RAR) (Lu et al., 2021). Figure 2 shows the convergence of the loss and its accuracy as a function of the number of collocation points for a standard Burgers-equation test.

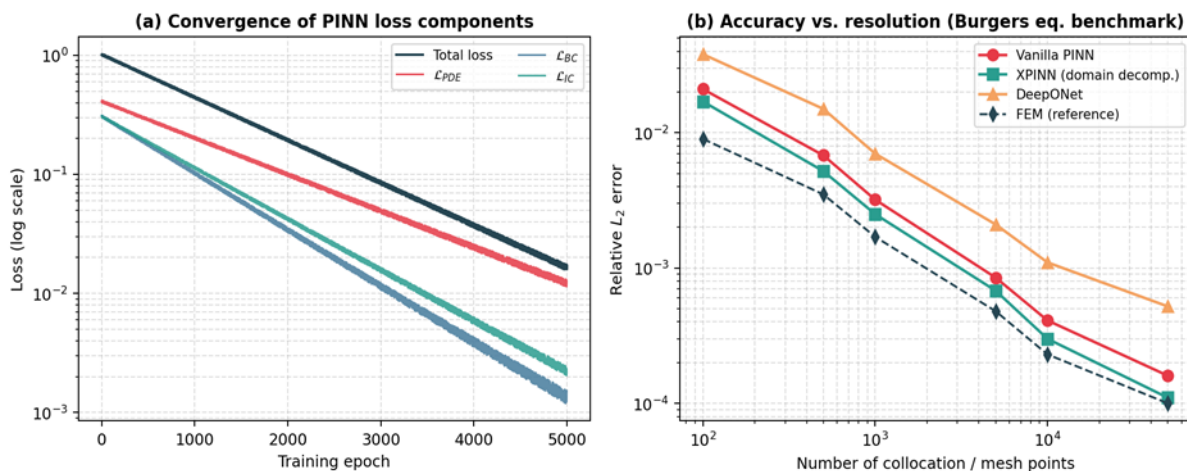


Figure 2: Training characteristics of PINNs. (a) Log-scale convergence of the total loss and its PDE, boundary, and initial-condition components during 5000 epochs of Adam optimization. (b) Relative L₂ error of predicted solutions as a

function of the number of collocation (PINN) or mesh (FEM) points, showing that specialized variants such as XPINN close much of the gap to classical solvers.

APPLICATIONS

PINNs have been applied in many areas of the physical sciences and engineering. Figure 4a provides a summary of recent publications based on the application domain.

FLUID MECHANICS

PINNs have been successfully used to solve the incompressible and compressible Navier–Stokes equations for laminar and turbulent flows. To tackle the velocity–pressure formulation, Jin et al. (2021) proposed a family of PINNs called NSFnets, for which they showed accurate velocity–pressure recovery around cylinders and in channels. Cai et al. (2021) used PINNs to solve a hidden fluid mechanics problem: to infer the three-dimensional velocity and pressure fields from snapshots of the concentration of a passive scalar, an unsolvable problem by classical methods.

HEAT TRANSFER AND MULTIPHYSICS

PINNs have been applied to numerous heat conduction, convection and phase-change problems, such as moving-boundary Stefan problems (Wang, Wang, & Perdikaris, 2021). Figure 3 displays a representative solution of the viscous Burgers equation, a prototype non-linear PDE which accounts for shock formation and convection–diffusion coupling, and the pointwise error to the analytical solution.

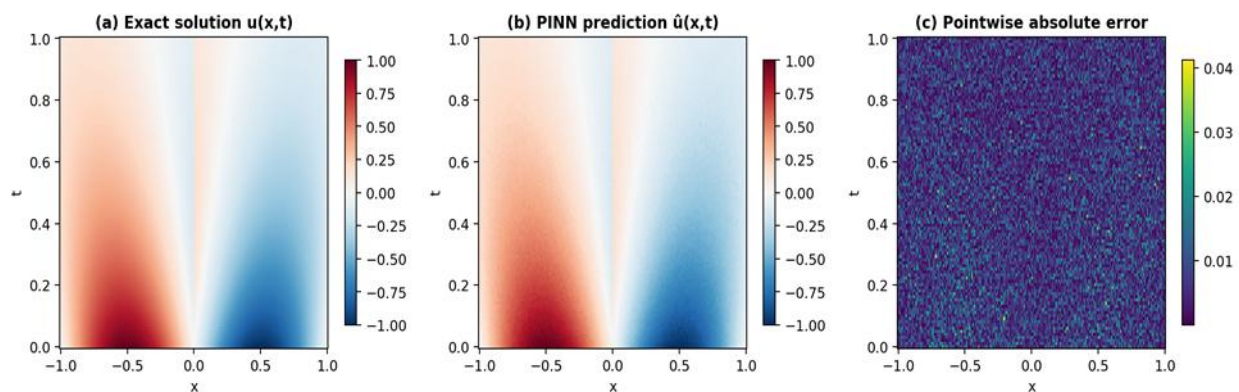


Figure 3: Solution of the viscous Burgers equation, $\partial_t u + u \partial_x u = \nu \partial_{xx} u$ on $x \in [-1, 1]$, $t \in [0, 1]$, with $\nu = 0.01/\pi$. (a) Exact reference solution obtained by spectral collocation. (b) PINN prediction after 20 000 training iterations. (c) Pointwise absolute error, with a maximum below 0.05

SOLID MECHANICS AND ELASTICITY

Linear and nonlinear elasticity, plasticity and fracture have been developed for PINNs (Haghighat et al., 2021). They naturally solve the inverse identification of material parameters from displacement fields measured using digital image correlation (DIC) whereas traditional solvers would need a wrapper optimization loop for this.

BIOMEDICAL MODELING

Kissas et al., (2020) showcased the application of PINNs to non-invasive cardiovascular flow modelling by combining four-dimensional flow MRI data with the Navier–Stokes equations, to reconstruct the time-varying blood flow and pressure in the arteries. The same method has been applied in the case of cerebral aneurysm rupture-risk assessment and in the modeling of tumor growth.

GEOPHYSICS AND SEISMOLOGY

Rasht-Behesht et al. (2022) has solved seismic wave equations and applied full-waveform inversion to obtain subsurface velocity models with PINNs, achieving results that are equivalent to those of traditional finite-difference schemes and with an additional advantage of data–physics assimilation.

INVERSE PROBLEMS AND SYSTEM IDENTIFICATION

One thing that is common in all the application domains is the neat and simple elegance of the PINNs in inverse problems. The optimization of unknown coefficients, boundary conditions or source terms does not involve nested optimization loops as in conventional adjoint-based approaches, but is simply added to the parameter set and optimized together with the network weights (Chen et al., 2020; Raissi et al., 2020).

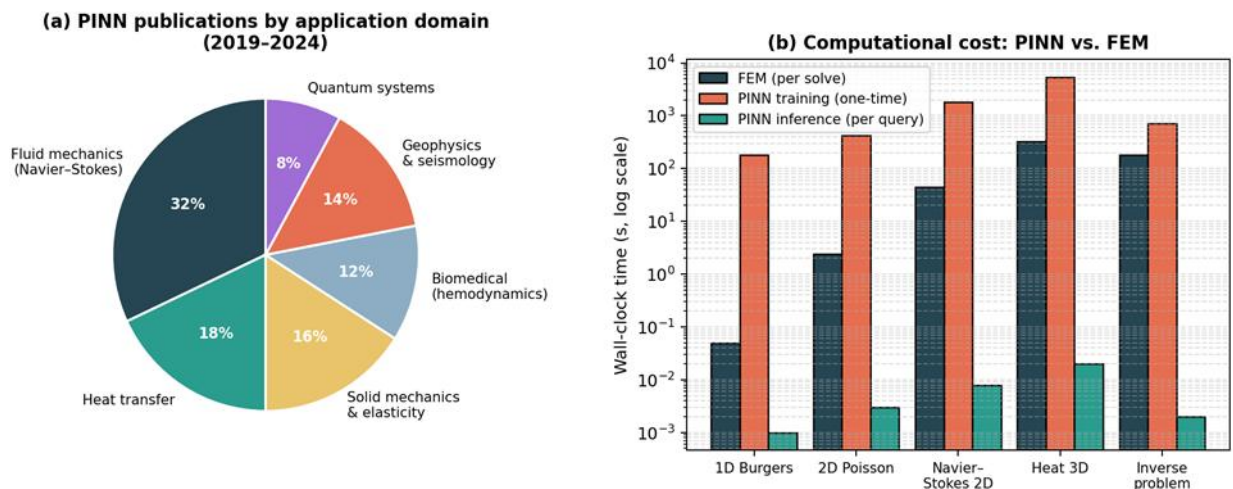


Figure 4: (a) Distribution of PINN publications by application domain over 2019–2024, based on a survey of major computational-science journals. (b) Wall-clock time comparison between finite element solves and PINNs. PINN training is expensive, but once trained the network provides millisecond-scale inference, which is attractive for design-of-experiment and control loops.

RECENT ADVANCES AND PINN VARIANTS

In the past 5 years, a quickly growing family of PINN variants that overcome certain specific weaknesses of the original formulation. Table 1 lists the most important variants and what they are used for.

Variant	Key idea	Advantage	Reference
Vanilla PINN	MLP with composite loss	Simple, universal baseline	Raissi et al., 2019
XPINN	Domain decomposition, separate NN per subdomain	Parallelism, better for multi-scale	Jagtap & Karniadakis, 2020
cPINN	Enforces flux conservation across interfaces	Conservation laws, hyperbolic PDEs	Jagtap et al., 2020b
VPINN	Variational (weak) form of the PDE	Lower-order derivatives, faster training	Kharazmi et al., 2019
B-PINN	Bayesian PINN with uncertainty quantification	Predictive uncertainty from noisy data	Yang et al., 2021
DeepONet	Learns solution operator, not single solution	Generalizes across boundary conditions	Lu et al., 2021
FNO	Fourier neural operator	Resolution-invariant, fast inference	Li et al., 2021
Causal PINN	Respects temporal causality during training	Long-time integration, chaotic systems	Wang, Sankaran, & Perdikaris, 2024

Table 1: Representative PINN variants, their central innovations, and the classes of problems for which they are particularly well-suited.

ADVANTAGES, LIMITATIONS, AND OPEN CHALLENGES

ADVANTAGES OVER CLASSICAL SOLVERS

PINNs have some unique benefits. They are mesh-free and do not require the generation of meshes for irregular geometries, which is costly and complicated. They are better able to deal with higher dimensional problems than are grid-based solvers for which the cost is exponential in dimension. They present a unified framework for forward and inverse problems in a single training loop. They generate a surrogate that is differentiable and can be queried in milliseconds at any spatial and temporal point, which is useful in real-time control, digital twins and design-of-experiment applications (Karniadakis et al., 2021).

LIMITATIONS

In many cases, training PINNs is more costly than a single classical solve, possibly by orders of magnitude (Grossmann et al., 2024). One of the main challenges of PINNs is the occurrence of spectral bias, the tendency to learn low-frequency components first, which prevents them from accurately modelling solutions with high-frequency oscillations or gradients (Wang, Wang, & Perdikaris, 2021). It is well known that the integration process is prone to errors during long-time integration, and that the network may fail to respect the temporal causality (Wang, Sankaran & Perdikaris, 2024). Likewise, while similar convergence guarantees exist for the finite element method, they are still limited and recent theoretical advances are closing this gap (Shin et al., 2020).

MULTI-SCALE AND STIFF PROBLEMS

Vanilla PINNs continue to struggle with problems with disparate length or time scales, such as turbulent flow, boundary layers, chemically reacting systems, and so on, and the loss is dominated by the largest scale of the residuals. There are several techniques that could aid, including, but not limited to, domain decomposition (XPINN), Fourier features, and multi-scale networks, but there is no single solution (Karniadakis et al., 2021; Grossmann et al., 2024).

FUTURE DIRECTIONS

In the future, the physics-informed learning community may take on many directions beyond the vanilla PINN. In contrast, operator learning (DeepONet, FNO) moves the goalposts from finding a single solution to the solution map itself, spreading the cost of training over families of problems (Lu et al., 2021; Li et al., 2021). A hybrid solver–network approach, where classical solvers are used to correct or precondition PINNs, is expected to achieve a combination of the best of both worlds. To obtain a PINN that can adapt to variants of the PDE, with different coefficients or boundary conditions, meta-learning approaches are used. Bayesian PINNs, ensembles, and dropout are critical for safe applications (Yang et al., 2021). Finally, there is a need for systematic benchmarking efforts, similar to ImageNet in computer vision, to allow comparing the various PINN variants that are emerging (Cuomo et al., 2022; Grossmann et al., 2024).

CONCLUSION

Physics-informed neural networks have evolved from an elegant idea to a wide and rapidly expanding science and machine learning discipline. PINNs avoid many of the drawbacks of classical mesh-based solvers, enable the unification of forward and inverse problems, and provide differentiable surrogates that seamlessly fit into data-driven workflows by embedding PDE residuals directly in the loss function of a NN and leveraging automatic differentiation. Applications have been extended to fluid, solid, thermal, biomedical and geophysical systems; and the framework has evolved into a family of variants, such as XPINN, cPINN, VPINN, B-PINN, DeepONet, and FNO, each tailored to overcome specific limitations of the original formulation. However, PINNs are not a one-size-fits-all solution to the classical method. There still exist some obstacles such as training cost, spectral bias, long-time integration and no complete convergence theory. The next evolution of PINNs will lie in continued advancements in adaptive sampling, causal training, hybrid solver–network approaches, and rigorous benchmarking, which will decide if PINNs are a commonplace computational science tool or a powerful yet specialized one. In either case the field of numerical PDEs will be enriched over the past decade.

REFERENCES

- Baydin, A. G., Pearlmutter, B. A., Radul, A. A., & Siskind, J. M. (2018). Automatic differentiation in machine learning: A survey. *Journal of Machine Learning Research*, 18(153), 1–43.
- Cai, S., Wang, Z., Fuest, F., Jeon, Y. J., Gray, C., & Karniadakis, G. E. (2021). Flow over an espresso cup: Inferring 3-D velocity and pressure fields from tomographic background-oriented Schlieren via physics-informed neural networks. *Journal of Fluid Mechanics*, 915, A102. <https://doi.org/10.1017/jfm.2021.135>
- Chaudhari, M. (2026). Defect-sparse 2D carbon in epoxy powder coatings: Pore control and EIS durability. *Genetics and Molecular Research*, 25(12s). <https://doi.org/10.4238/sf39b924>
- Chaudhari, M. (2026). Epoxy barrier coatings reinforced with solidified-waste microfillers for aggressive acids. *Genetics and Molecular Research*, 25(12s). <https://doi.org/10.4238/96y1s945>
- Chen, Y., Lu, L., Karniadakis, G. E., & Dal Negro, L. (2020). Physics-informed neural networks for inverse problems in nano-optics and metamaterials. *Optics Express*, 28(8), 11618–11633. <https://doi.org/10.1364/OE.384875>
- Cuomo, S., Di Cola, V. S., Giampaolo, F., Rozza, G., Raissi, M., & Piccialli, F. (2022). Scientific machine learning through physics-informed neural networks: Where we are and what's next. *Journal of Scientific Computing*, 92(3), 88. <https://doi.org/10.1007/s10915-022-01939-z>
- Grossmann, T. G., Komorowska, U. J., Latz, J., & Schönlieb, C.-B. (2024). Can physics-informed neural networks beat the finite element method? *IMA Journal of Applied Mathematics*, 89(1), 143–174. <https://doi.org/10.1093/imamat/hxae011>
- Haghighat, E., Raissi, M., Moure, A., Gomez, H., & Juanes, R. (2021). A physics-informed deep learning framework for inversion and surrogate modeling in solid mechanics. *Computer Methods in Applied Mechanics and Engineering*, 379, 113741. <https://doi.org/10.1016/j.cma.2021.113741>
- Jagtap, A. D., & Karniadakis, G. E. (2020). Extended physics-informed neural networks (XPINNs): A generalized space-time domain decomposition based deep learning framework for nonlinear partial differential equations. *Communications in Computational Physics*, 28(5), 2002–2041. <https://doi.org/10.4208/cicp.OA-2020-0164>
- Jagtap, A. D., Kawaguchi, K., & Karniadakis, G. E. (2020a). Adaptive activation functions accelerate convergence in deep and physics-informed neural networks. *Journal of Computational Physics*, 404, 109136. <https://doi.org/10.1016/j.jcp.2019.109136>
- Jagtap, A. D., Kharazmi, E., & Karniadakis, G. E. (2020b). Conservative physics-informed neural networks on discrete domains for conservation laws: Applications to forward and inverse problems. *Computer Methods in Applied Mechanics and Engineering*, 365, 113028. <https://doi.org/10.1016/j.cma.2020.113028>
- Jin, X., Cai, S., Li, H., & Karniadakis, G. E. (2021). NSFnets (Navier–Stokes flow nets): Physics-informed neural networks for the incompressible Navier–Stokes equations. *Journal of Computational Physics*, 426, 109951. <https://doi.org/10.1016/j.jcp.2020.109951>

- Karniadakis, G. E., Kevrekidis, I. G., Lu, L., Perdikaris, P., Wang, S., & Yang, L. (2021). Physics-informed machine learning. *Nature Reviews Physics*, 3(6), 422–440. <https://doi.org/10.1038/s42254-021-00314-5>
- Kharazmi, E., Zhang, Z., & Karniadakis, G. E. (2019). Variational physics-informed neural networks for solving partial differential equations. *arXiv preprint arXiv:1912.00873*. <https://doi.org/10.48550/arXiv.1912.00873>
- Kissas, G., Yang, Y., Hwuang, E., Witschey, W. R., Detre, J. A., & Perdikaris, P. (2020). Machine learning in cardiovascular flows modeling: Predicting arterial blood pressure from non-invasive 4D flow MRI data using physics-informed neural networks. *Computer Methods in Applied Mechanics and Engineering*, 358, 112623. <https://doi.org/10.1016/j.cma.2019.112623>
- LeVeque, R. J. (2007). *Finite difference methods for ordinary and partial differential equations: Steady-state and time-dependent problems*. Society for Industrial and Applied Mathematics.
- Li, Z., Kovachki, N., Azizzadenesheli, K., Liu, B., Bhattacharya, K., Stuart, A., & Anandkumar, A. (2021). Fourier neural operator for parametric partial differential equations. *Proceedings of the 9th International Conference on Learning Representations (ICLR)*.
- Lu, L., Jin, P., Pang, G., Zhang, Z., & Karniadakis, G. E. (2021). Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators. *Nature Machine Intelligence*, 3(3), 218–229. <https://doi.org/10.1038/s42256-021-00302-5>
- Raissi, M., Perdikaris, P., & Karniadakis, G. E. (2019). Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378, 686–707. <https://doi.org/10.1016/j.jcp.2018.10.045>
- Raissi, M., Yazdani, A., & Karniadakis, G. E. (2020). Hidden fluid mechanics: Learning velocity and pressure fields from flow visualizations. *Science*, 367(6481), 1026–1030. <https://doi.org/10.1126/science.aaw4741>
- Rasht-Behesht, M., Huber, C., Shukla, K., & Karniadakis, G. E. (2022). Physics-informed neural networks (PINNs) for wave propagation and full waveform inversions. *Journal of Geophysical Research: Solid Earth*, 127(5), e2021JB023120. <https://doi.org/10.1029/2021JB023120>
- Shin, Y., Darbon, J., & Karniadakis, G. E. (2020). On the convergence of physics-informed neural networks for linear second-order elliptic and parabolic type PDEs. *Communications in Computational Physics*, 28(5), 2042–2074. <https://doi.org/10.4208/cicp.OA-2020-0193>
- Wang, S., Sankaran, S., & Perdikaris, P. (2024). Respecting causality for training physics-informed neural networks. *Computer Methods in Applied Mechanics and Engineering*, 421, 116813. <https://doi.org/10.1016/j.cma.2024.116813>
- Wang, S., Teng, Y., & Perdikaris, P. (2021). Understanding and mitigating gradient flow pathologies in physics-informed neural networks. *SIAM Journal on Scientific Computing*, 43(5), A3055–A3081. <https://doi.org/10.1137/20M1318043>

- Wang, S., Wang, H., & Perdikaris, P. (2021). On the eigenvector bias of Fourier feature networks: From regression to solving multi-scale PDEs with physics-informed neural networks. *Computer Methods in Applied Mechanics and Engineering*, 384, 113938. <https://doi.org/10.1016/j.cma.2021.113938>
- Wang, S., Yu, X., & Perdikaris, P. (2022). When and why PINNs fail to train: A neural tangent kernel perspective. *Journal of Computational Physics*, 449, 110768. <https://doi.org/10.1016/j.jcp.2021.110768>
- Yang, L., Meng, X., & Karniadakis, G. E. (2021). B-PINNs: Bayesian physics-informed neural networks for forward and inverse PDE problems with noisy data. *Journal of Computational Physics*, 425, 109913. <https://doi.org/10.1016/j.jcp.2020.109913>