

---

# Fast PINN Eigensolvers via Biconvex Reformulation

---

Akshay Sai Banderwaar

School of Mechanical Sciences

Indian Institute of Technology, Goa

banderwaar.sai.22063@iitgoa.ac.in

Abhishek Gupta

School of Mechanical Sciences

Indian Institute of Technology, Goa

abhishek@iitgoa.ac.in

## Abstract

Eigenvalue problems have a distinctive forward-inverse structure and are fundamental to characterizing a system’s thermal response, stability, and natural modes. Physics-Informed Neural Networks (PINNs) offer a mesh-free alternative for solving such problems but are often orders of magnitude slower than classical numerical schemes. In this paper, we introduce a reformulated PINN approach that casts the search for eigenpairs as a biconvex optimization problem, enabling fast and provably convergent alternating convex search (ACS) over eigenvalues and eigenfunctions using analytically optimal updates. Numerical experiments show that PINN-ACS attains high accuracy with convergence speeds up to  $500\times$  faster than gradient-based PINN training. We release our codes at [https://github.com/NeurIPS-ML4PS-2025/PINN\\_ACSS\\_CODES](https://github.com/NeurIPS-ML4PS-2025/PINN_ACSS_CODES).

## 1 Introduction

Solving eigenvalue problems for differential operators involves finding non-trivial eigenvalue–eigenfunction pairs (eigenpairs) where the operator scales the function by its eigenvalue. Determining the function for a given eigenvalue resembles a forward problem, while extracting the eigenvalue from a function mimics an inverse problem. Physics-Informed Neural Networks (PINNs) have shown promise as high-resolution, mesh-free solvers for forward and inverse problems individually [1, 2], and these strengths could be brought to bear on the uniquely coupled forward–inverse structure of eigenvalue problems as well. However, existing PINN algorithms are known to be orders of magnitude slower than conventional numerical schemes [3, 4], primarily due to the difficulty of optimizing the highly non-convex PINN loss function. This loss combines terms representing the underlying differential equation and the initial and/or boundary conditions, which are often of incommensurable scale and can conflict under finite network capacity [5]. Moreover, the initial flatness of the network’s output—a symptom of the vanishing gradient problem [6]—tends to draw PINNs towards trivial solutions [7, 8], an issue that is especially deleterious for eigenvalue problems.

Recent PINN methods for discovering eigenpairs rely on standard gradient-based iterative optimization [9, 10, 11, 12], and therefore remain highly susceptible to conflicting loss terms and spurious basins of attraction in non-convex loss landscapes [13, 14]. One way to mitigate the conflict is to enforce boundary conditions directly into the network’s output, but applications have largely been limited to Dirichlet boundaries [11, 12]. To reinforce the mesh-free flexibility of PINNs with accelerated convergence, this paper introduces a simple yet powerful trick: reformulating the PINN loss as a biconvex optimization problem. The key idea is to restrict training to the linear output layer of the PINN, keeping its hidden layers fixed to values initialized either randomly or according to some heuristic [15]. For linear differential operators, this makes the physics residual linear in terms of the network’s unknown parameters [16, 17]. Consequently, the highly non-convex loss landscape is transformed into a biconvex one: convex (linear least-squares) in terms of the parameterized eigenfunction when the eigenvalue is fixed, and vice versa. We show that this structure admits a theoretically convergent alternating convex search (ACS) over eigenpairs, where each convex

subproblem is solved optimally via analytical solutions. The overall framework not only yields substantial speedups compared to gradient-based optimization, but also improves accuracy through the analytic updates. The main contributions of this work are summarized below.

- The differential eigenvalue problem is recast as a biconvex PINN optimization. Unlike prior PINN approaches, our reformulation accommodates varied boundary conditions and even applies to cases where the eigenvalue appears explicitly in the boundary operator (e.g., Euler’s buckling of a column with a free end).
- Alternating convex search for PINN optimization is proposed. Starting at a random initial eigenvalue estimate, ACS iteratively fixes one variable (the eigenvalue or the eigenfunction) and updates the other via analytic results. PINN-ACS converges rapidly and can be parallelized over a population of estimates for simultaneous discovery of multiple eigenpairs.
- We demonstrate the performance of our method on Euler’s buckling problem, for solving Helmholtz equation on an L-shaped domain, and for obtaining the natural frequencies of a thin plate (governed by a fourth-order biharmonic equation). PINN-ACS achieves speedups exceeding  $10\times$ , and up to  $500\times$ , compared to gradient-based PINN training.

## 2 Mathematical Preliminaries

The general form of an eigenvalue problem for a differential operator defined over domain  $\Omega$ , with boundary  $\partial\Omega$ , can be written as:

$$\mathcal{D}(u_i)(\mathbf{x}) + \lambda_i h(u_i)(\mathbf{x}) = 0, \quad \mathbf{x} \in \Omega, \quad (1a)$$

$$\mathcal{B}(u_i)(\mathbf{x}) = 0, \quad \mathbf{x} \in \partial\Omega. \quad (1b)$$

Here,  $\mathcal{D}(\cdot)$  is the differential operator,  $h(\cdot)$  is a function,  $\mathbf{x}$  represents the spatial coordinate,  $\mathcal{B}(\cdot)$  is the boundary operator,  $\lambda_i$  is the  $i^{\text{th}}$  eigenvalue and  $u_i(\mathbf{x})$  is the corresponding eigenfunction. In this work, we focus on linear self-adjoint operators, a common setting in recent studies and across many scientific applications [9, 18]. Under this condition, the eigenfunctions are mutually orthogonal:

$$\langle u_i, u_j \rangle = \int_{\Omega} u_i(\mathbf{x}) u_j(\mathbf{x}) d\mathbf{x} = 0, \text{ for } i \neq j. \quad (2)$$

## 3 Methodology

### 3.1 PINN Loss for Eigenvalue Problems

Consider a multilayer perceptron parameterized by  $\theta$  to approximate the  $i^{\text{th}}$  eigenfunction as  $u_i(\mathbf{x}; \theta)$ . The PINN loss  $\mathcal{L}(\lambda_i, \theta)$  in terms of the model parameters and eigenvalue  $\lambda_i$  can then be defined as:

$$\begin{aligned} \arg \min_{\lambda_i, \theta} \mathcal{L}(\lambda_i, \theta) &= \int_{\Omega} |\mathcal{D}(u_i(\mathbf{x}; \theta)) + \lambda_i h(u_i(\mathbf{x}; \theta))|^2 d\Omega + \alpha_{\text{BC}} \int_{\partial\Omega} \mathcal{B}(u_i(\mathbf{x}; \theta))^2 d(\partial\Omega) \\ &\quad + \alpha_{\text{ref}} (u_i(\mathbf{x}^{\text{ref}}; \theta) - u_{\text{ref}})^2 + \alpha_{\text{ortho}} \sum_{j=1}^{i-1} \left( \int_{\Omega} u_i(\mathbf{x}; \theta) u_j(\mathbf{x}) d\Omega \right)^2. \end{aligned} \quad (3)$$

The first term captures the error in satisfying the differential equation; the second term enforces the boundary conditions; the third term prevents a trivial solution by prescribing  $u_{\text{ref}} \neq 0$  at some reference point  $\mathbf{x}^{\text{ref}}$ ; the final term enforces orthogonality to eigenfunctions already discovered;  $\alpha_{\text{BC}}$ ,  $\alpha_{\text{ref}}$ ,  $\alpha_{\text{ortho}}$  control the trade-offs between components. The loss is usually evaluated numerically by computing the integrand at randomly sampled collocation points and summing the results.

An alternate way to achieve solution non-triviality is to substitute the third term by a normalization condition:  $\int_{\Omega} u_i(\mathbf{x}; \theta)^2 d\Omega = 1$ . However, this would induce nonlinear interactions between network parameters, hampering the biconvexification enabled by the form of Eq. (3), discussed next.

### 3.2 Biconvex Reformulation

In a depth- $L$  neural net,  $\theta = \{W^{[l]}, \mathbf{b}^{[l]}\}_{l=1}^L$  and the transformation performed at the  $l^{\text{th}}$  layer is  $a^{[l]}(\mathbf{x}) = \sigma^{[l]}(W^{[l]} a^{[l-1]}(\mathbf{x}) + \mathbf{b}^{[l]})$ , where  $a^{[0]}(\mathbf{x}) = \mathbf{x}$ . While  $\sigma^{[L]}$  is typically set to the identity

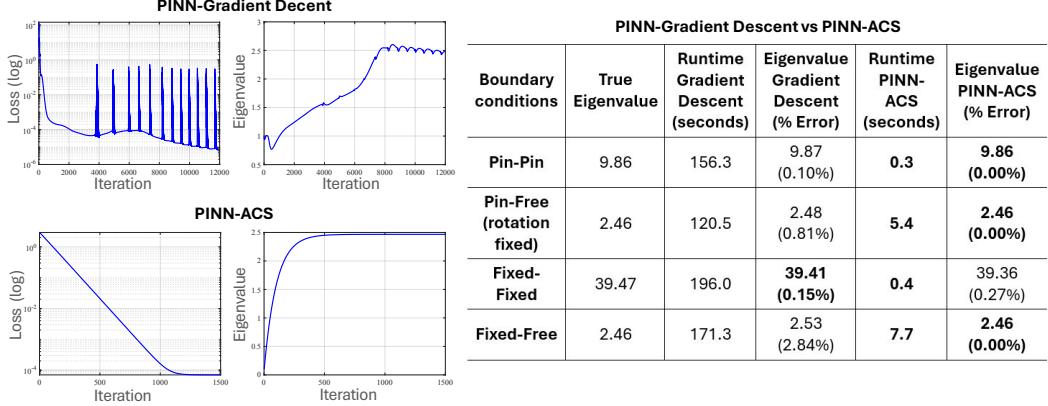


Figure 1: Illustrative plots for Euler’s buckling show that PINN-ACS leads to monotonic convergence as predicted by Theorem 1. Gradient descent using Adam [10, 20] oscillates due to conflicting loss terms. Tabulated average runtimes indicate substantial speedups with ACS alongside higher accuracy.

function and  $b^{[L]} = \mathbf{0}$ , the other non-linear activations,  $\sigma^{[l]}$ , render the network’s output non-linear in terms of the hidden layers’ weights and biases. This turns Eq. (3) into a mathematical program with a highly non-convex loss landscape, posing stiff optimization challenges unique to PINNs [14, 19].

In this paper, we reformulate the PINN eigensolver by restricting training to just the network’s output layer. Hidden layers are either set to random values or according to some heuristic (e.g., by transferring weights from a related source problem) and are left untrained. Randomization preserves a network’s universal approximation capacity under sufficient width and appropriate selection of random parameter distributions [21, 22]. As a result, we can write  $u_i(\mathbf{x}; \theta)$  as  $u_i(\mathbf{x}; W^{[L]})$  which is linear in terms of  $W^{[L]}$ . Plugging this simplified form of the eigenfunction into Eq. (3) gives  $\mathcal{L}(\lambda_i, W^{[L]})$  where the third and fourth terms turn into linear least-squares expressions in  $W^{[L]}$ . Note that this would not be the case if the normalization condition was employed (as a substitute to the third term) for achieving non-trivial solutions. Although the first and (in certain cases) second terms give rise to non-linear interactions between  $W^{[L]}$  and  $\lambda_i$ , conditioning them on a fixed  $\lambda_i$  reduces the overall loss to a convex linear least-squares problem in terms of  $W^{[L]}$  alone. Likewise, conditioning the loss on a fixed  $W^{[L]}$  reduces it to a convex linear least-squares form in  $\lambda_i$ . The loss function is therefore “biconvexified” when jointly considering  $\lambda_i, W^{[L]}$ . Each convex subproblem admits a closed-form analytical solution by the Moore-Penrose pseudoinverse when the coefficients of  $W^{[L]}$  or  $\lambda_i$  are assembled into design matrices. In case of ill-conditioned design matrices, the closed-form solution obtained under Tikhonov regularization of  $W^{[L]}$  provides a numerically stable approximation to the pseudoinverse (which is usually computed via singular value decomposition in most numerical libraries), and converges to it as the regularization coefficient approaches zero.

### 3.3 Alternating Convex Search

Alternating convex search (ACS) is a robust algorithm for biconvex optimization [23]. In the context of the eigenvalue problem, each convex subproblem possesses an optimal analytical solution as discussed above. The alternating iterations—where one variable (either the eigenvalue or the parameterized eigenfunction) is fixed to its value from the previous step while the other is updated—are thus theoretically convergent and, in practice, converge rapidly.

**Theorem 1.** Let  $\mathcal{L}^{(t)}$  be the loss function value for the differential eigenvalue problem after the  $t^{\text{th}}$  ACS iteration. Then the sequence  $\{\mathcal{L}^{(t)}\}_{t \in \mathbb{N}}$  generated by ACS converges monotonically.

*Proof.* Squared terms in the loss function given by Eq. (3) imply that its value must be greater than or equal to 0. Analytically optimal updates in each ACS iteration guarantee that  $\mathcal{L}^{(t)}$  is monotonically decreasing. Since  $\mathcal{L}$  is bounded from below, the sequence must converge to a limit value.

Empirical analysis shows that ACS typically converges to the eigenpair closest to its initial random eigenvalue estimate. By initializing a population of estimates, multiple ACS strands can be run in

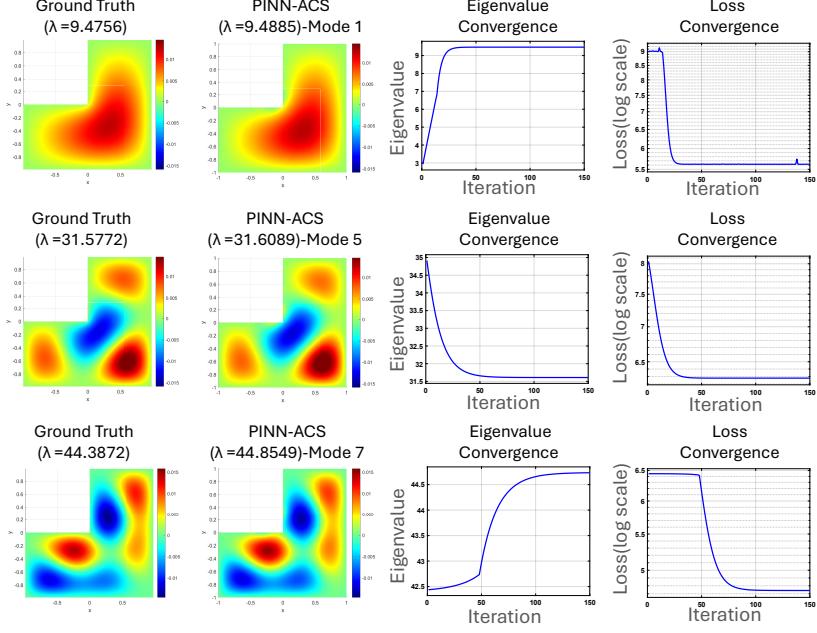


Figure 2: PINN-ACS closely approximates the ground truth (fine-mesh finite difference solutions) for the Laplace operator. Gradient descent results are omitted due to convergence difficulties within a reasonable time, especially for higher-order modes, consistent with observations in [10]. In contrast, PINN-ACS achieves monotonic loss convergence in agreement with theory.

parallel to discover several eigenpairs simultaneously. The resulting eigenfunctions are incorporated into the fourth term of Eq. (3) for the next generation of population-based search, thereby guiding each generation toward distinct eigenpairs. Given distributed hardware, this approach substantially reduces the time required to identify all eigenpairs within prescribed search bounds for the eigenvalues.

## 4 Numerical Results

A shallow neural net with 500 neurons suffices for the 1D and 2D test problems considered in this paper. A shallow architecture allows all required derivatives of the network’s output with respect to its inputs to be expressed analytically [24], thereby enabling efficient computation of higher-order derivatives without relying on automatic differentiation tools. Random Fourier features (cosine activations) were employed for effective learning of high frequency eigenfunctions [25]. Appropriate selection of the randomization bandwidth is important, with larger bandwidths facilitating the learning of high frequency patterns [7]. This selection could be automated by adapting the method from [26]. All codes for this work were developed in MATLAB R2023b taking advantage of its fast matrix computation capabilities, without GPU acceleration. The codes were run on a single 11th Gen Intel(R) Core(TM) i5-1135G7 processor with 8 GB RAM  $\sim$ 2.40 GHz.

### 4.1 Euler’s Buckling

The transverse deflection  $w$  of a column under axial load is given by the fourth-order equation:

$$\frac{d^4w}{dx^4} + \lambda^2 \frac{d^2w}{dx^2} = 0, \quad x \in [0, 1]. \quad (4)$$

The first eigenvalue of this problem produces critical buckling loads under varied boundary conditions, including unique cases where the eigenvalue appears in the boundary operator; e.g., zero shear condition at a free end is imposed by setting  $\frac{d^3w}{dx^3} + \lambda^2 \frac{dw}{dx} = 0$ . Results comparing the performance of gradient-based PINN training using the formulation in [10] and PINN-ACS are presented in Fig. (1). The key takeaway is that PINN-ACS is at least  $10\times$  faster than gradient descent. For the pin-pin and fixed-fixed cases, it achieves  $500\times$  speedup, while maintaining generally higher accuracy. The hidden model parameters were sampled uniformly at random from the interval  $[-1, 1]$  for this example.

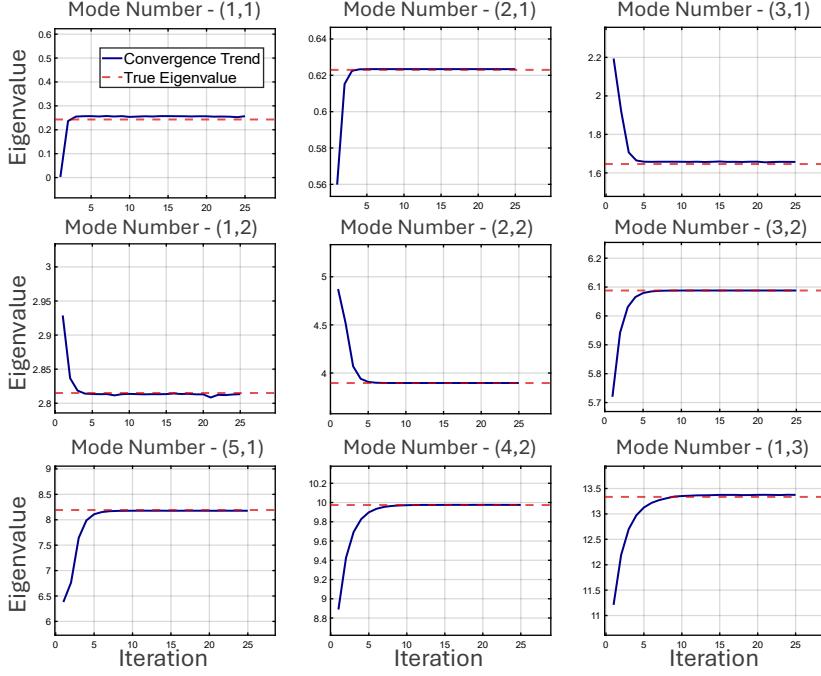


Figure 3: Convergence trends of PINN-ACS eigenvalues to the ground truth for free vibrations of a thin plate [27]. Gradient descent results excluded as it fails to converge reliably for higher modes.

#### 4.2 Helmholtz Equation on L-shaped Domain

The Helmholtz equation is the eigenvalue problem for the Laplace operator and is stated as:

$$\frac{\partial^2 w(x, y)}{\partial x^2} + \frac{\partial^2 w(x, y)}{\partial y^2} = -\lambda w(x, y), \quad (x, y) \in \Omega. \quad (5)$$

The problem is solved on an L-shaped domain with Dirichlet boundaries, i.e.,  $w(x, y) = 0, (x, y) \in \partial\Omega$ . Comparisons to a classical numerical scheme are shown in Fig. (2), with PINN-ACS achieving a high degree of accuracy for both low- and high-frequency eigenfunctions. The hidden parameters of the model were sampled uniformly at random from the interval  $[-8\pi, 8\pi]$  for this example.

#### 4.3 Biharmonic Equation for Thin Plates

The vibration of thin plates is governed by the following biharmonic eigenvalue problem:

$$\frac{\partial^4 w}{\partial x^4} + 2 \frac{\partial^4 w}{\partial x^2 \partial y^2} + \frac{\partial^4 w}{\partial y^4} = \lambda^4 w, \quad (x, y) \in [0, 10] \times [0, 5]. \quad (6)$$

The problem is solved for a rectangular plate under simply supported boundary conditions, assuming Poisson's ratio  $\nu = 0.3$ . The PINN's hidden parameters were sampled uniformly from the interval  $[-\pi, \pi]$ . Fig. (3) depicts the smooth convergence of the estimated eigenvalues to the ground truth.

## 5 Conclusions

This paper fills a gap in PINN applications by introducing a fast PINN eigensolver based on a biconvex loss reformulation. The alternating convex search (ACS) procedure is theoretically guaranteed to achieve monotonic convergence for biconvex problems, and provides substantial speedups in practice compared to gradient-based PINN training. Future work will extend the approach to nonlinear operators and larger-scale physics applications. One way to address nonlinearity is by the Picard method [28] where the problem is iteratively linearized at the differential equation level using previous solution approximations, thereby preserving applicability of the PINN-ACS algorithm on alternating linear least-squares subproblems. More broadly, ACS provides a general framework for solving inverse problems where both unknown field functions and parameters are simultaneously sought.

## Acknowledgements

This work was supported by the Ramanujan Fellowship from the Anusandhan National Research Foundation, Government of India (Grant No. RJF/2022/000115).

## References

- [1] Maziar Raissi, Paris Perdikaris, and George E Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational physics*, 378:686–707, 2019.
- [2] Juan Diego Toscano, Vivek Oommen, Alan John Varghese, Zongren Zou, Nazanin Ahmadi Daryakenari, Chenxi Wu, and George Em Karniadakis. From pinns to pikans: Recent advances in physics-informed machine learning. *Machine Learning for Computational Science and Engineering*, 1(1):1–43, 2025.
- [3] Nick McGreivy and Ammar Hakim. Weak baselines and reporting biases lead to overoptimism in machine learning for fluid-related partial differential equations. *Nature machine intelligence*, 6(10):1256–1269, 2024.
- [4] Tamara G Grossmann, Urszula Julia Komorowska, Jonas Latz, and Carola-Bibiane Schönlieb. Can physics-informed neural networks beat the finite element method? *IMA Journal of Applied Mathematics*, 89(1):143–174, 2024.
- [5] Rafael Bischof and Michael A Kraus. Multi-objective loss balancing for physics-informed deep learning. *Computer Methods in Applied Mechanics and Engineering*, 439:117914, 2025.
- [6] Sifan Wang, Yujun Teng, and Paris Perdikaris. Understanding and mitigating gradient flow pathologies in physics-informed neural networks. *SIAM Journal on Scientific Computing*, 43(5):A3055–A3081, 2021.
- [7] Jian Cheng Wong, Chin Chun Ooi, Abhishek Gupta, and Yew-Soon Ong. Learning in sinusoidal spaces with physics-informed neural networks. *IEEE Transactions on Artificial Intelligence*, 5(3):985–1000, 2022.
- [8] Franz Martin Rohrhofer, Stefan Posch, Clemens Gößnitzer, and Bernhard Geiger. On the role of fixed points of dynamical systems in training physics-informed neural networks. *Transactions on Machine Learning Research*, 2023(1):490, 2023.
- [9] Ido Ben-Shaul, Leah Bar, Dalia Fishelov, and Nir Sochen. Deep learning solution of the eigenvalue problem for differential operators. *Neural Computation*, 35(6):1100–1134, 2023.
- [10] Seongjoon Yoo, Minseo Kang, Heonjun Yoon, and Taejin Kim. A physics-informed neural network approach for solving structural eigenvalue problem. *International Journal of Precision Engineering and Manufacturing*, pages 1–14, 2025.
- [11] Conor Rowan, John Evans, Kurt Maute, and Alireza Doostan. Solving engineering eigenvalue problems with neural networks using the rayleigh quotient. *arXiv preprint arXiv:2506.04375*, 2025.
- [12] Julian Fernandez Bonder and Ariel M Salort. A pinns approach for the computation of eigenvalues in elliptic problems. *arXiv preprint arXiv:2507.03126*, 2025.
- [13] Aditi Krishnapriyan, Amir Gholami, Shandian Zhe, Robert Kirby, and Michael W Mahoney. Characterizing possible failure modes in physics-informed neural networks. *Advances in neural information processing systems*, 34:26548–26560, 2021.
- [14] Jian Cheng Wong, Abhishek Gupta, Chin Chun Ooi, Pao-Hsiung Chiu, Jiao Liu, and Yew-Soon Ong. Evolutionary optimization of physics-informed neural networks: Evo-pinn frontiers and opportunities. *arXiv preprint arXiv:2501.06572*, 2025.
- [15] Claudio Gallicchio and Simone Scardapane. Deep randomized neural networks. In *Recent Trends in Learning From Data: Tutorials from the INNS Big Data and Deep Learning Conference (INNSBDDL2019)*, pages 43–68. Springer, 2020.

- [16] Suchuan Dong and Yiran Wang. A method for computing inverse parametric pde problems with random-weight neural networks. *Journal of Computational Physics*, 489:112263, 2023.
- [17] Sifan Wang, Bowen Li, Yuhan Chen, and Paris Perdikaris. Piratenets: Physics-informed deep learning with residual adaptive networks. *Journal of Machine Learning Research*, 25(402):1–51, 2024.
- [18] Rishi Mishra, Ganapathy Krishnamurthi, Balaji Srinivasan, and Sundararajan Natarajan. Eigpielm: A mesh-free approach for efficient eigen-analysis with physics-informed extreme learning machines. *arXiv preprint arXiv:2508.15343*, 2025.
- [19] Nicholas Sung, Jian Cheng Wong, Chin Chun Ooi, Abhishek Gupta, Pao-Hsiung Chiu, and Yew-Soon Ong. Neuroevolution of physics-informed neural nets: Benchmark problems and comparative results. In *Proceedings of the Companion Conference on Genetic and Evolutionary Computation*, pages 2144–2151, 2023.
- [20] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [21] Boris Igelnik and Yoh-Han Pao. Stochastic choice of basis functions in adaptive function approximation and the functional-link net. *IEEE transactions on Neural Networks*, 6(6):1320–1329, 1995.
- [22] Ming Li, Sho Sonoda, Feilong Cao, Yu Guang Wang, and Jiye Liang. How powerful are shallow neural networks with bandlimited random weights? In *International Conference on Machine Learning*, pages 19960–19981. PMLR, 2023.
- [23] Jochen Gorski, Frank Pfeuffer, and Kathrin Klamroth. Biconvex sets and optimization with biconvex functions: a survey and extensions. *Mathematical methods of operations research*, 66(3):373–407, 2007.
- [24] 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.
- [25] Matthew Tancik, Pratul Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. *Advances in neural information processing systems*, 33:7537–7547, 2020.
- [26] Suchuan Dong and Jielin Yang. On computing the hyperparameter of extreme learning machines: Algorithm and application to computational pdes, and comparison with classical and high-order finite elements. *Journal of Computational Physics*, 463:111290, 2022.
- [27] Stephen Timoshenko and Sergius Woinowsky-Krieger. Theory of plates and shells. 1959.
- [28] Claudio Paniconi and Mario Putti. A comparison of picard and newton iteration in the numerical solution of multidimensional variably saturated flow problems. *Water Resources Research*, 30(12):3357–3374, 1994.