Evolutionary Optimization of Physics-Informed Neural Networks: Evo-PINN Frontiers and Opportunities

Jian Cheng Wong[1], Abhishek Gupta[2], Chin Chun Ooi[1], Pao-Hsiung Chiu[1], Jiao Liu[3], and Yew-Soon Ong[1,3]
[1]Agency for Science, Technology and Research (A*STAR), 138634, Singapore
[2]Indian Institute of Technology (IIT) Goa, Ponda, 403401, India
[3]Nanyang Technological University, 639798, Singapore

***Abstract*—Deep learning models trained on finite data lack a complete understanding of the physical world. On the other hand, physics-informed neural networks (PINNs) are infused with such knowledge through the incorporation of mathematically expressible laws of nature into their training loss function. By complying with physical laws, PINNs provide advantages over purely data-driven models in limited-data regimes and present as a promising route towards Physical AI. This feature has propelled them to the forefront of scientific machine learning, a domain characterized by scarce and costly data. However, the vision of accurate physics-informed learning comes with significant challenges. This work examines PINNs in terms of model optimization and generalization, shedding light on the need for new algorithmic advances to overcome issues pertaining to the training speed, precision, and generalizability of today's PINN models. Of particular interest are gradient-free evolutionary algorithms (EAs) for optimizing the uniquely complex loss landscapes arising in PINN training. Methods synergizing gradient descent and EAs for discovering bespoke neural architectures and balancing multiple terms in physics-informed learning objectives are positioned as important avenues for future research. Another exciting track is to cast EAs as a meta-learner of generalizable PINN models. To substantiate these proposed avenues, we further highlight results from recent literature to showcase the early success of such approaches in addressing the aforementioned challenges in PINN optimization and generalization.**

***Index Terms*—Physics-informed neural networks, scientific machine learning, evolutionary optimization, neuroevolution, generalizability.**

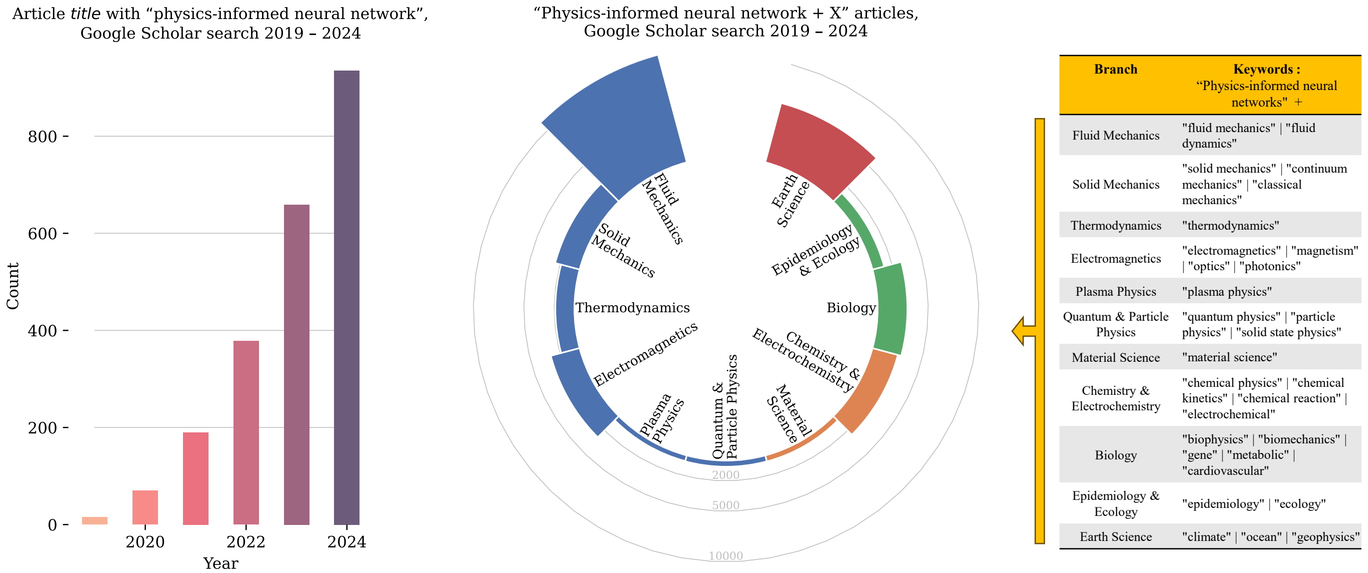
I. Introduction

OVER the past decade, advances in deep learning have garnered increasing attention for their potential to address pressing challenges in science and engineering. The emerging topic of artificial intelligence (AI) for science promises to transform technological frontiers [1]. For example, learned models have shown the capacity to accelerate the design and discovery of novel materials, photonic devices, proteins, drugs, and renewable energy technologies such as wind turbines. However, unlike conventional applications of AI in vision, speech, or natural language processing where there is a deluge of available data, applications in scientific domains are often characterized by extreme data scarcity. This has raised concerns regarding the veracity of such models when trained only on small data. The cost of acquiring quality data gives rise to a practical bottleneck, leading to naive statistical models which sometimes generate physically inconsistent outputs. These shortcomings become even more consequential in the context of Agentic AI systems [2–4], which are designed to autonomously drive scientific discovery and decision-making. In such systems, a lack of embedded physical understanding by models can lead to flawed reasoning or unsafe actions when interfacing with the real world [5, 6].

The shortcomings of standard data-driven models have seeded the field of *scientific machine learning* where minimizing deviations from the known laws of nature serves to boost model precision and data efficiency [21, 22]. Associated methods offer unique pathways to *rationalizable* AI, defined in [23], that is cognizant of fundamental physics principles. A myriad of approaches to instil physics consistency into learned models can be crafted, with the most suitable methods being naturally dependent on the task at hand and the type of knowledge available [24–27]. Some instantiations include the design of model architectures that preserve invariances or symmetries representative of the underlying system [28], data augmentation with theory-based synthetic data (e.g., first principles numerical simulations) [29], or introduction of learning biases into the training process [30].

This effort focuses on the growing body of literature on *physics-informed neural networks* (PINNs), popularized in recent years by the work of Raissi *et al.* [31]. The essential idea of a PINN—*where mathematically-defined learning biases are directly incorporated into a neural network's loss function*—is simultaneously decades old, with its earlier work in the 1990s [32, 33], and also a breath of fresh air for AI. This physics-informed loss is amenable to various forms of scientific knowledge and theories, including conservation laws and/or governing equations in the form of ordinary and partial differential equations (ODEs and PDEs), yet retains immense flexibility in the choice of network architectures (e.g., multilayer perceptron, convolutional or graph neural networks) and learning algorithms (e.g., gradient-based or gradient-free). This makes it possible for scientific knowledge uncovered over centuries, from the micro-scale evolution of quantum-mechanical systems to the macro-scale behavior of solids, fluids, and propagation of electromagnetic fields, to be incorporated into the learned models of today, thereby compensating for a lack of labeled data.

Corresponding author: Jian Cheng Wong (e-mail: wongj@a-star.edu.sg).



<table>
<tr><th>Branch</th><th>Keywords :
“Physics-informed neural networks" +</th></tr>
<tr><td>Fluid Mechanics</td><td>"fluid mechanics" | "fluid dynamics"</td></tr>
<tr><td>Solid Mechanics</td><td>"solid mechanics" | "continuum mechanics" | "classical mechanics"</td></tr>
<tr><td>Thermodynamics</td><td>"thermodynamics"</td></tr>
<tr><td>Electromagnetics</td><td>"electromagnetics" | "magnetism" | "optics" | "photonics"</td></tr>
<tr><td>Plasma Physics</td><td>"plasma physics"</td></tr>
<tr><td>Quantum & Particle Physics</td><td>"quantum physics" | "particle physics" | "solid state physics"</td></tr>
<tr><td>Material Science</td><td>"material science"</td></tr>
<tr><td>Chemistry & Electrochemistry</td><td>"chemical physics" | "chemical kinetics" | "chemical reaction" | "electrochemical"</td></tr>
<tr><td>Biology</td><td>"biophysics" | "biomechanics" | "gene" | "metabolic" | "cardiovascular"</td></tr>
<tr><td>Epidemiology & Ecology</td><td>"epidemiology" | "ecology"</td></tr>
<tr><td>Earth Science</td><td>"climate" | "ocean" | "geophysics"</td></tr>
</table>

Fig. 1: Google Scholar search of the literature shows the popularity of PINN over the years between 2019 and 2024, and PINN studies in diverse scientific areas. *Data collected in November 2024.*

TABLE I: Potential applications of PINNs across various scientific domains

Branch	Applications
Fluid Mechanics	Turbomachinery design [7], airfoil design [8]
Solid Mechanics	Topology optimization [9, 10], vehicle crashworthiness design [11], additive manufacturing [12, 13], digital twin [14, 15]
Chemistry & Electrochemistry	Battery management systems [16]
Quantum & Particle Physics	Diffuse optical tomography [17], nuclear fusion physics analysis [18, 19]
Material Science	Composite-tool systems during manufacture [20]

The potential of such physics-informed models has spawned significant interest across various scientific domains. Trends from a Google Scholar search of the literature in the last six years (2019-2024) are depicted in Fig. 1. The figure reveals rapid growth in the popularity of PINNs in different branches of science, with many demonstrations being developed using open software libraries such as JAX [34], NVIDIA PhysicsNeMo [35], SciANN [36], and DeepXDE [37].

Despite the broad applicability of PINNs (ref. to Table I for some potential applications), the switch from standard data-driven loss functions to physics-informed learning objectives has brought unforeseen difficulties in optimizing the latter’s uniquely complex landscape [38–40]. These training difficulties undermine the theoretical advantages of PINNs, and are currently a barrier to their adoption in place of established numerical methods in science and engineering. Prior surveys [27, 41–45] review PINN methods, applications, limitations, trends, and prospects from interdisciplinary viewpoints, but a common point of emphasis is that efficient optimization of PINNs remains an open problem [44].

Separately, despite the physics-informed loss, the physical validity of a PINN is limited to scenarios encountered during training, even after a complicated training process. Most PINNs are designed to model a specific PDE instance, and model training can be highly sensitive to choice of architecture and optimization hyperparameters. Consequently, reusing identical hyperparameter configurations to train new PINN models within the same PDE family often fails to produce satisfactory results. They do not inherently generalize learned physics to new scenarios beyond their training regime—e.g., changing PDE parameters, initial conditions or boundary conditions—thereby continuing to suffer from generalizability issues similar to standard data-driven learning algorithms.

In light of the above, this paper examines PINNs through the lens of their optimization and generalization challenges. As evidenced by Fig. 1, there is a vast amount of innovative PINN literature, however, this paper specifically focuses on a curated subset that best illustrates both these optimization and generalization challenges, and the exciting opportunities for novel, targeted algorithms and methods. Notably, most PINN implementations today make use of gradient-based optimization algorithms. The ubiquity of the method may be attributed to inertia from the primacy of stochastic gradient descent (SGD) in conventional data-rich deep learning tasks. However, SGD is no silver bullet. Relative to standard deep learning, physics-informed loss functions are not only highly non-convex but also exhibit disparities in the magnitudes of back-propagated gradient signals between the data and physics loss terms (see III-A1). These issues lead to pathologies that hinder effective training [46, 47]. The resulting optimization trajectory is littered with spurious minima, making it extremely challenging to find globally optimum solutions that satisfy physics. There is a high chance for gradient descent to get trapped in a local minimum that fails to satisfy the requisite physics, a possibility discussed in greater detail in Section III.

Fig. 2 provides a schematic summary of some of the main

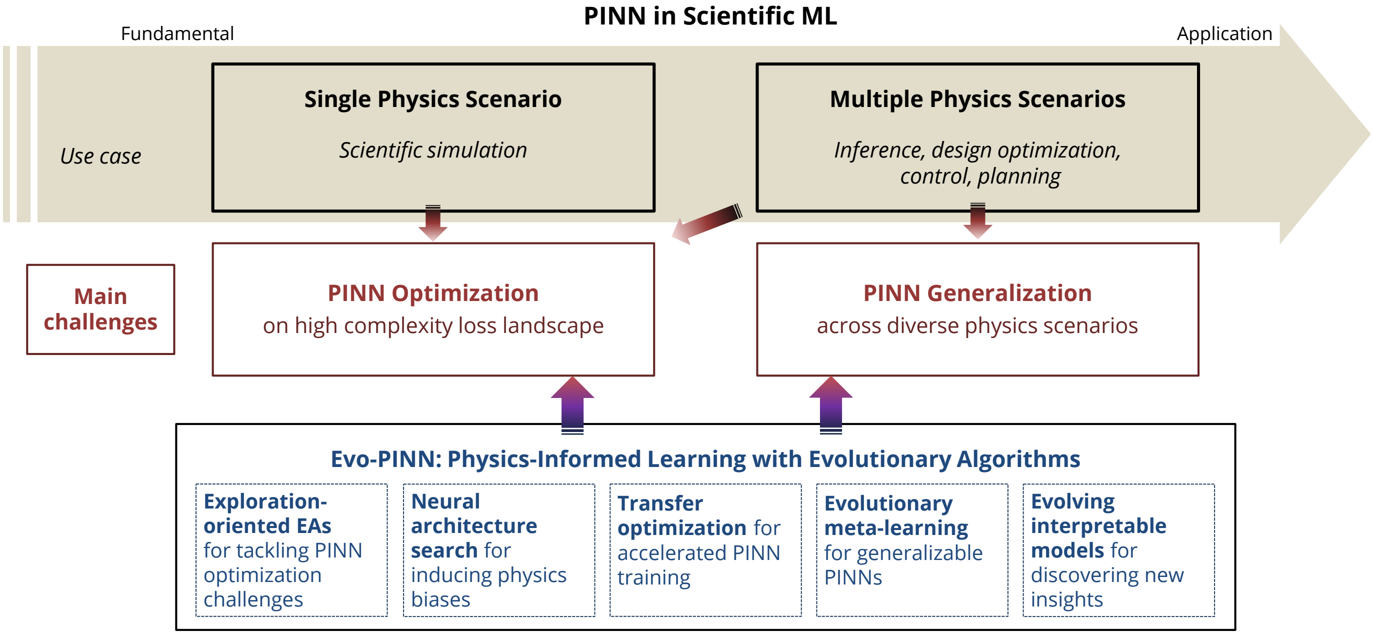

PINN in Scientific ML
Fundamental
Application
Use case
Single Physics Scenario
Scientific simulation
Multiple Physics Scenarios
Inference, design optimization, control, planning
Main challenges
PINN Optimization
on high complexity loss landscape
PINN Generalization
across diverse physics scenarios
Evo-PINN: Physics-Informed Learning with Evolutionary Algorithms
Exploration-oriented EAs for tackling PINN optimization challenges
Neural architecture search for inducing physics biases
Transfer optimization for accelerated PINN training
Evolutionary meta-learning for generalizable PINNs
Evolving interpretable models for discovering new insights

Fig. 2: Boosting scientific ML with evolutionary optimization of physics-informed neural networks (Evo-PINN).

computational challenges inhibiting the vision of physics-informed machine learning towards AI for Science, and some corresponding proposed solutions. In Section IV, arguments are presented in support of our main claim: that algorithmic ideas developed over the years within the computational intelligence community may be uniquely well-suited for physics-informed learning. Of particular interest are the gradient-free evolutionary algorithms (EAs) that suffer less from spurious local minima due to their exploration-oriented global search capacity [48, 49]. An associated research direction is the synergy of EAs with gradient descent (in the spirit of memetic algorithms), thereby marrying the best of both worlds. While population-based EAs may be advantageous in navigating highly non-convex loss landscapes and balancing trade-offs between conflicting objectives, local search via gradient descent can facilitate rapid convergence to precise solutions.

In addition to optimizing network parameters, EAs enable the discovery of bespoke architectures for physics-informed learning [50]. Evolutionary mechanisms could also prove useful for guiding training data generation in areas with higher errors on governing physics equations, having shown evidence of improved convergence to correct solutions [51]. Accurate solutions to novel PDEs can be rapidly obtained through evolutionary transfer optimization, leveraging multiple knowledge sources during optimization for specific target tasks [48]. Another exciting track is to cast EAs as a meta-learner of generalizable PINN models [52]. This may be achieved through a bilevel formulation where an upper-level EA searches for PINN configurations that can be quickly fine-tuned for various downstream physics tasks at the lower-level.

The advancement of Evo-PINNs is expected to have far-reaching impact on establishing PINN methodologies at the forefront of scientific machine learning. However, there are only a handful of related works in the literature, implying huge scope for future research and development.

II. Overview of PINN Methodologies

PINNs are designed to learn a mapping between the input and output variables while satisfying specified mathematical constraints that represent the physical phenomenon or dynamical process of interest [31]. Given their ubiquity in modern science and engineering, PDEs are one of the most studied physics priors integrated in PINNs. Without loss of generality, we consider a physics-informed learning problem with input variables x (spatial dimension) and t (time dimension), and an output variable u satisfying PDEs of the general form:

$$\begin{aligned}
&\text{PDE:} && \mathcal{N}_{\vartheta}[u(x,t)] = h(x,t), && x \in \Omega, t \in (0,T] && \text{(1a)}\\
&\text{IC:} && u(x,t=0) = u_0(x), && x \in \Omega && \text{(1b)}\\
&\text{BC:} && \mathcal{B}[u(x,t)] = g(x,t), && x \in \partial\Omega, t \in (0,T] && \text{(1c)}
\end{aligned}$$

where the general differential operator $\mathcal{N}_{\vartheta}[u(x,t)]$ contains PDE parameters ϑ and can include linear and/or nonlinear combinations of temporal and spatial derivatives of u, and $h(x,t)$ is an arbitrary source term in the domain $x \in \Omega, t \in (0,T]$. Equation 1b specifies the initial condition (IC), $u_0(x)$, at time t=0, and Equation 1c specifies the boundary condition (BC) such that $\mathcal{B}[u(x,t)]$ equates to $g(x,t)$ at the domain boundary $\partial\Omega$, where $\mathcal{B}[\cdot]$ can either be an identity operator (Dirichlet BC), a differential operator (Neumann BC), or a mixed identity-differential operator (Robin BC).

A. Physics-Informed Loss

PINNs are distinct from DNNs in that additional physics priors are incorporated as an essential component of its physics-informed loss function:

$$\mathcal{L}_{\text{PINN}} = \lambda_{physics}\ \mathcal{L}_{physics} + \lambda_{data}\ \mathcal{L}_{data}, \qquad (2)$$

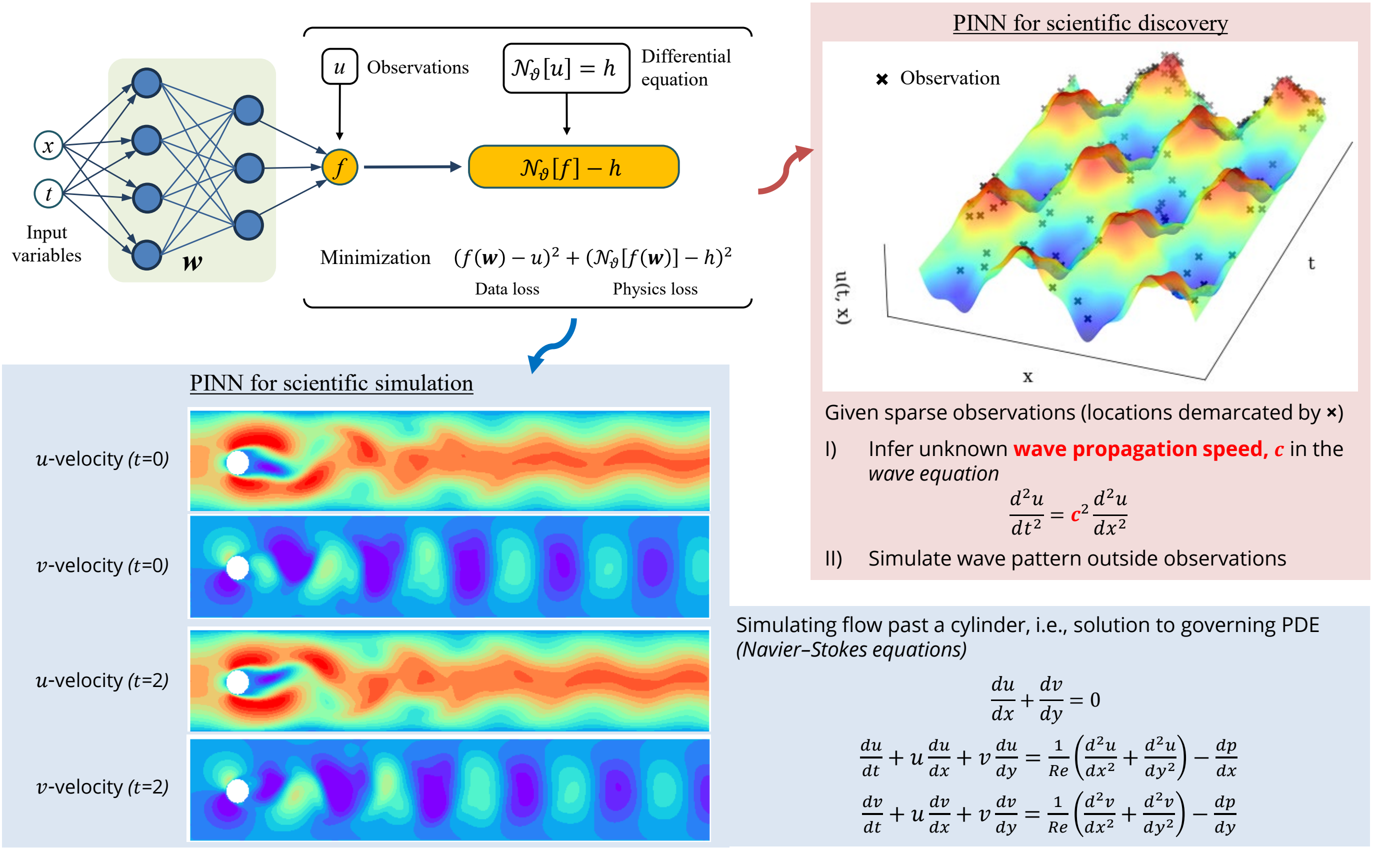

x
t
Input variables
$\boldsymbol{w}$
u Observations
$\mathcal{N}_\vartheta[u] = h$ Differential equation
f
$\mathcal{N}_\vartheta[f] - h$
Minimization $(f(\boldsymbol{w}) - u)^2 + (\mathcal{N}_\vartheta[f(\boldsymbol{w})] - h)^2$
Data loss
Physics loss
PINN for scientific discovery
× Observation
u(t, x)
t
x
Given sparse observations (locations demarcated by ×)
I) Infer unknown wave propagation speed, c in the wave equation
$\frac{d^2u}{dt^2} = c^2 \frac{d^2u}{dx^2}$
II) Simulate wave pattern outside observations
PINN for scientific simulation
u-velocity (t=0)
v-velocity (t=0)
u-velocity (t=2)
v-velocity (t=2)
Simulating flow past a cylinder, i.e., solution to governing PDE (Navier–Stokes equations)
$\frac{du}{dx} + \frac{dv}{dy} = 0$
$\frac{du}{dt} + u\frac{du}{dx} + v\frac{du}{dy} = \frac{1}{Re}\left(\frac{d^2u}{dx^2} + \frac{d^2u}{dy^2}\right) - \frac{dp}{dx}$
$\frac{dv}{dt} + u\frac{dv}{dx} + v\frac{dv}{dy} = \frac{1}{Re}\left(\frac{d^2v}{dx^2} + \frac{d^2v}{dy^2}\right) - \frac{dp}{dy}$

Fig. 3: PINN handles both scientific simulation and scientific discovery. Example applications are: 1) accurate simulation of transient vortex shedding behind a cylinder obtained by learning a PINN for a single problem solely using the governing physics (2D transient N-S equations); and 2) inference of the wave propagation speed, c, given sparse observations (locations demarcated by $\times$), by learning an inference PINN model with a partially specified 1D transient wave equation.

where the relative weights $\lambda_{physics}$ and λ_{data} control the component-wise trade-off. The physics-informed component can consist of multiple PDE, IC, and BC loss terms:

$$\mathcal{L}_{physics} = \lambda_{pde}\ \mathcal{L}_{pde} + \lambda_{ic}\ \mathcal{L}_{ic} + \lambda_{bc}\ \mathcal{L}_{bc} \tag{3a}$$

$$\mathcal{L}_{pde} = \|\mathcal{N}_\vartheta[f(\cdot;\boldsymbol{w})] - h(\cdot)\|^2_{L^2(\Omega\times(0,T])} \tag{3b}$$

$$\mathcal{L}_{ic} = \|f(\cdot, t=0;\boldsymbol{w}) - u_0(\cdot)\|^2_{L^2(\Omega)} \tag{3c}$$

$$\mathcal{L}_{bc} = \|\mathcal{B}[f(\cdot;\boldsymbol{w})] - g(\cdot)\|^2_{L^2(\partial\Omega\times(0,T])} \tag{3d}$$

where $f(\cdot;\boldsymbol{w})$ is the PINN output function with network parameters $\boldsymbol{w}$, evaluated on a set of label-free training points (collocation points) sampled from the respective spatio-temporal domain. An illustrative schematic of a PINN model is presented in Fig. 3. Evaluation of differential operators, e.g., $\frac{d\ f(\cdot;\boldsymbol{w})}{dt}$, $\frac{d\ f(\cdot;\boldsymbol{w})}{dx}$, and $\frac{d^2 f(\cdot;\boldsymbol{w})}{dx^2}$, are generally required for computation of the physics-informed loss on these training samples. They can be conveniently computed by automatic differentiation [31] and/or numerical differentiation [53, 54] methods at any sample location.

Fundamentally, PINNs arrive at an accurate and physics-compliant prediction for a desired physics scenario by forcing its outputs $f(\cdot;\boldsymbol{w})$ to satisfy Equation 1 through minimizing the physics-informed loss (Equation 3). The effectiveness of the optimization of network parameters $\boldsymbol{w}$ through the physics-informed loss is a primary determinant of the accuracy of the PINN model. In contrast, the typical data-driven loss (as common in a DNN) computes the MSE between the model prediction $f(x_i, t_i;\boldsymbol{w})$ and available target u_i^{label} over i=1, ..., n labeled samples:

$$\mathcal{L}_{\text{DNN}} = \mathcal{L}_{data} = \frac{1}{n}\sum_{i=1}^{n}\left(u_i^{label} - f(x_i, t_i;\boldsymbol{w})\right)^2 \tag{4}$$

B. *Physics-Informed Learning Advantages*

The advantages of physics-informed learning are demonstrated in Fig. 4. Given only a few labeled data (n=5), a DNN model (with sufficient representation capacity) remains unable to correctly reconstruct the entire system, although it fits all training samples well. In contrast, a PINN model can accurately learn the solution from the same set of labeled data. By enforcing physics constraints on a set of *label-free* training samples (n_{pde}=1000) uniformly drawn from the computation domain utilizing the aforementioned PINN loss, a PINN can learn the solution even without any labeled data. The governing physics in this example is the convection-diffusion equation $\alpha\frac{du}{dx} - \frac{d^2u}{dx^2} = 0$ with PDE parameter α, a ubiquitous physics model that describes the distribution of a scalar quantity (e.g., mass, energy, temperature) in the presence of convective transport and diffusion [55].

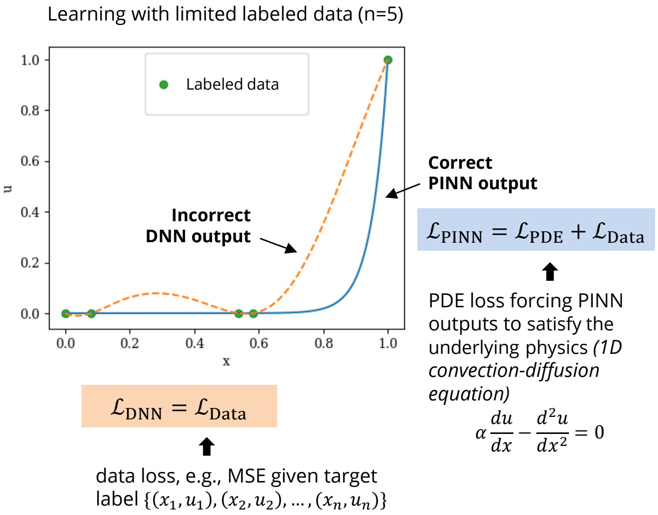


Fig. 4: Illustrative example showing how a PINN model can make accurate prediction by learning with physics laws (1D convection-diffusion equation) and few labeled data, whereas a conventional DNN model fails to learn the correct output.

C. PINN for Scientific Simulation

PDEs offer concise mathematical descriptions of diverse physics phenomena in nature. With PINNs, it is possible to simulate an entire physical system by training the model to approximate the solution of the PDEs specific to the target physics, without any labeled data (i.e., solely using the physics-informed loss). A good example of this is the fluid dynamic simulation of vortex shedding behind a cylinder depicted in Fig. 3, which was obtained by using a PINN to solve the transient 2D Navier-Stokes (N-S) equations [56].

Compared to classical numerical solvers employing the finite difference, finite element, or finite volume method [57], PINNs have the significant advantage of being mesh-free. Meshing, which is the process of domain discretization, is an essential, yet non-trivial task in classical numerical methods. Numerical methods can fail to find the correct solution if the meshing is not appropriately done. On the other hand, PINNs can provide a solution without similar mesh requirements. Additionally, the solution obtained through PINN is differentiable, and its compact representation requires lower memory demand for storage. This makes PINNs a promising alternative to traditional numerical methods for simulating physics phenomena; an extensive review is presented in [41].

While most PINN studies address single physics problem, a PINN model can also incorporate additional input parameters, such as geometry, Reynolds number, or material properties [58–60]. These models, also referred to as parameterized PINN, can then provide predictions across a set of physics scenarios, rather than just one.

D. PINN for Scientific Discovery

In addition to directly simulating physical behavior, PINNs can be seamlessly extended to various hypothesis inference problems common in scientific discovery. These problems typically involve an incomplete and potentially noisy set of observed data points, and the goal is to infer information about the system through optimizing the hypothesis space. The loss function used consists of both physics-informed (to capture relevant physics) and data (observations) components (Equation 2). The hypothesis spaces of the inference problem are included as optimization variables along with the network parameters during training; see example in Fig. 3.

The PINN models are uniquely suited for hypothesis inference when compared to numerical solvers and can effectively synergize theory and observation data from various sources such as sensors and imaging, even for ill-posed PDE problems. It can be combined with probabilistic inference methods (e.g., Bayesian-PINN [61]), with the probabilistic framework offering a mathematical tool to handle uncertainties arising from both noisy data and physics assumptions [61, 62].

In current literature, PINN models have been applied to: *1.) Inference of system-level parameters* such as the underlying governing equations, constitutive relationships or system parameters (e.g., Reynolds number, wave speed) based on observations [63–67]. *2.) Inference of unobserved field variables* across the entire domain based on limited observations, e.g., predicting pressure from velocity in particle imaging velocimetry [68]. Many have employed PINNs towards the inference of domain field variables in various problems spanning fluid dynamics [69], heat transfer [70], and biomedical sciences [71, 72]. *3.) PDE-constrained design optimization* towards a set of desired (observation) outcomes. The PINN is used as an adjoint optimizer, enabling the design parameters to be optimized towards the desired outcome while fulfilling all specified relevant physics during training. This has been successfully demonstrated on topology optimization [73] and metamaterial [74] design problems.

III. Optimization and Generalization Challenges

However, PINN's advantages come at the cost of increased difficulty in optimizing PINN model parameters, with many early papers showing long training times and limited predictive performance on more complex problems [31, 38–40]. Unsuccessful PINN training (premature convergence) can lead to unacceptable (unphysical) simulation outcomes. Moreover, parameterized PINNs can still face issues with generalizability since the accuracy of interpolation-based predictions on a new problem is not guaranteed, and training a well-generalizable PINN model may not be an easy task. In this section, we shed light on some contributing factors to the optimization and generalization challenges—two pre-dominant hurdles to the wide-spread adoption of PINNs in real-world applications today—and how they may originate from fundamental limitations in gradient-based approaches.

A. Challenges in Optimizing PINN Models

In many studies involving PINN, it was observed that the training iterations required for convergence far exceed those needed for typical data-driven DNN. We relate the optimization difficulty of PINNs to a few factors below.

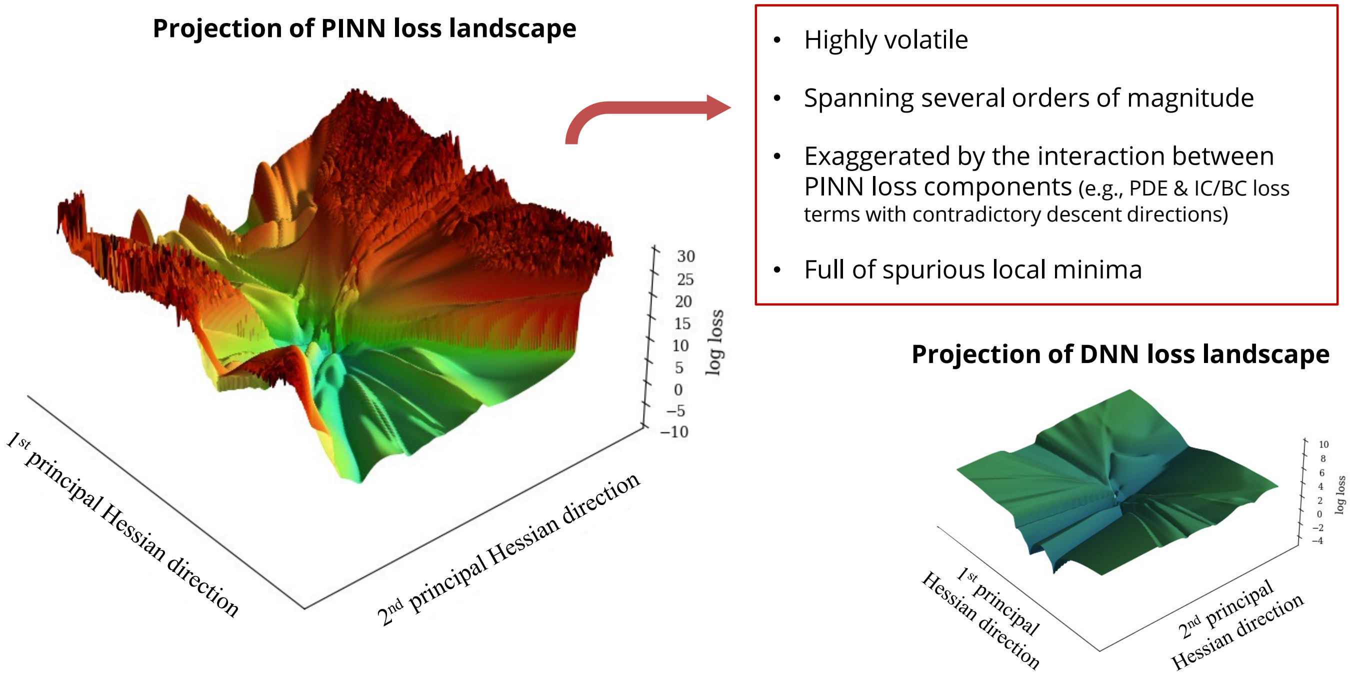


Fig. 5: The plots contrast the local loss landscapes of PINN and DNN, for the convection-diffusion problem presented in Subsection II-B. After 100,000 training iterations, we project their corresponding loss values onto the plane spanned by the first two principal Hessian directions as per previously reported literature [38] to get an intuition for the loss landscape.

1) Complex Loss Landscape: The physics-informed loss usually consists of multiple PDE and IC/BC loss terms (Equation 3), each with distinct scales and characteristics during PINN training. For instance, the scale of PDE loss can span several orders of magnitude depending on the underlying differential operators and the complexity of the solution, and this range is only revealed during training. Additionally, some loss terms change more dramatically than others during training, as shown by prior analyses through the lens of neural tangent kernel (NTK) [75, 76]. These characteristics make the proper scaling of different loss terms challenging.

In addition, the PINN optimization landscapes are littered with spurious minima and are more complex than those of usual data-driven DNN models. Fig. 5 illustrates the loss landscapes of PINN and DNN models for the convection-diffusion problem presented in Subsection II-B. To investigate the topological differences in their loss landscapes, we first computed the Hessians of their respective loss functions (i.e., $\mathcal{L}_{\text{PINN}}$ or $\mathcal{L}_{\text{DNN}}$) with respect to the model parameters: $\mathbf{H} = \nabla^2\ \mathcal{L}(\boldsymbol{w})$. We then extracted the leading two eigenvectors and eigenvalues of the Hessian (i.e., first two principal Hessian directions) which correspond to the directions of highest curvature in the loss landscape. By perturbing the trained models along these dominant eigenvectors and evaluating the resulting loss values, we can obtain a more informative characterization of the local topology than perturbing the model parameters in random directions [38]. This visualization provides evidence of a much higher degree of complexity and ruggedness of the loss landscape for PINN compared to usual data-driven DNN models. The smoothness of the latter is conducive to gradient-based algorithms that can easily find descent paths from the point of initialization to a good local optimum that fits the training data well. This outcome corroborates the primacy of SGD and its variants in today's deep learning tasks. In contrast, the contradictory descent directions from different loss terms encountered during PINN training can adversely impact optimization [77, 78]. It is worth noting that when the loss terms are purely physics-based (e.g., PDE and IC/BC), they are not theoretically contradictory. However, during the optimization process, especially when the solution is still far from the true optimum, the local gradient directions of these loss components can still become practically conflicting. This issue arises due to the combined effects of neural network expressivity, the structure of the loss landscape (e.g., scales and convergence dynamics), and the limitations of optimizers, which can cause certain objectives to dominate or suppress others during early or intermediate stages of training.

Although these powerful gradient-based algorithms are inseparable from the recent success of deep learning, they are no silver bullet. For instance, gradient-based methods are susceptible to being trapped in spurious local minima, which is often the case in the context of PINN optimization. Typical deep learning applications do not require a fully converged solution, as they prioritize generalization performance over training precision. However, this is not the case for PINNs, where reaching the global minimum is essential to ensure the model's fidelity to the underlying physics. A small violation of the PDE constraints can result in an unphysical PINN solution while a small violation of the BC or IC constraints corresponds to an entirely different PDE problem [79, 80].

2) Initial Local Minimum Trap and Spectral Bias: PINN models employing standard activation functions (such as *tanh*, *sigmoid*) when initialized using common techniques (such as the Xavier method) tend to produce flat output functions at initialization, making them immediately susceptible to a local minimum trap whereby the PINN loss already sees a minimized PDE loss term. This theoretical insight, discussed in [81–83], applies across a range of physics scenarios where the underlying PDE is trivially satisfied by flat functions. A strong bias around flat functions can therefore be difficult for gradient descent to escape, especially when the true solution, that must simultaneously minimize *all* PDE and IC/BC loss terms, is not accessible near the point of initialization.

Another closely related issue stems from the known saturation property of activation functions commonly used for PINN. Such saturation property can cause backpropagated gradients to vanish whenever the network weights are shifted towards a larger distribution, as driven by the need to approximate higher frequencies and steep gradients common in physics. This makes further gradient descent of the network weights difficult, adversely causing a learning bias towards smooth functions (lower frequencies are learned better than higher frequencies). This known tendency—also called *spectral bias*—has been studied through the lens of NTK [84, 85], revealing a significant discrepancy in convergence rates for different frequency components in the output during training. While this has been regarded as a natural regularizer with potential benefits to the generalization performance of DNNs, the consequences are highly deleterious for PINNs. Spectral bias makes it difficult for PINNs to accurately learn complex nonlinear physics behaviors [86], e.g., in the vicinity of steep gradients characterizing shock waves. These local discrepancies further deter propagation of the correct initial and boundary conditions into the computation domain, which has been suggested to be critical for successful PINN training [80, 87].

This saturation also translates to increased difficulties for gradient-based algorithms to alter the behavior of PINN output far away from the input origin. As a result, PINN struggles to learn over long time horizons, partly because the learning bias towards smooth functions can cause the gradient descent to prematurely converge to a trivial solution (satisfying the PDE) before the initial and boundary conditions can properly propagate to a later time [88]. An associated tendency for gradient-based PINN training to be drawn towards unstable fixed point solutions of dynamical systems was analyzed in [89], with experiments suggesting that they form strong basins of attraction in the PINN loss landscape.

3) Sensitivity to Sample Size and Location: The training samples of the PINN model are typically uniformly drawn from the input domain and are equally important everywhere due to their physical correspondence with the spatio-temporal extent of the problem. While the use of mini-batch samples is a standard procedure in gradient-based training, it remains an open question how one can efficiently reduce the batch size (and hence the training cost) while maintaining the desired PINN model accuracy [90]. Using a large batch size is crucial for the accuracy of PINNs due to the potential need to learn high frequencies or complex structures in the *a priori* unknown target function [91]. However, this inevitably increases training costs. Furthermore, oversampling on a high loss region may adversely slow convergence since it can be harder to learn such local complex gradient behaviour—a cause of stiffness during training [39, 40, 54]. On the other hand, an overly small batch size will result in noisy (big jump in loss values) and inconsistent local gradient signals [92], thereby causing issues during gradient descent.

4) Current Research Directions: Since the seminal work of Raissi *et al.* [31], architecture design and training methods for PINNs have continued to evolve. Recent advancements often combine multiple learning strategies to tackle the aforementioned PINN optimization issues [93–95]. *Curriculum learning* improves convergence by initiating training on a better-conditioned, simpler version of the original problem and then systematically advancing to target problem [38]. Physics-informed learning often succumbs to premature convergence for challenging problems, however, the same PDE equation with altered parameters or IC/BCs may be a simpler problem. In this scenario, one can start training the PINN model on the easier problem and then gradually vary the problem during training to approach the target problem. This approach assumes that each training step brings the model closer to the optimal solution of the next (harder) problem, thereby improving the next round of training. Hence, curriculum learning can be viewed as an initialization process to guide the model toward the desired solution utilizing *a priori* understanding of how the system behaves as its problem parameters change. *Adaptive loss-balancing strategy* can also be considered as a form of curriculum learning, where the training is initiated with the loss weights corresponding to a simpler optimization problem. It involves assigning appropriate weights to balance different terms in the PINN loss functions [47, 75–78], and is regarded as an important technique to improve convergence.

Recent studies have proposed enhancements to PINN learning by leveraging network architectures that incorporate Fourier feature embeddings, skip connections, and random weight factorization [93–95]. PINNs with skip connections and periodic activation functions retain a greater proportion of gradients during backpropagation from output to input, thereby mitigating spectral bias caused by activation saturation [81, 83, 96]. Adaptive techniques that adjust the slope of activation functions have also been introduced to further control saturation and enhance training efficiency [97, 98]. These methods embed inductive biases—such as high-frequency behavior characteristic of certain PDEs—directly into the network architecture. Additionally, physics-informed cell representations have been developed to address spectral bias away from the input origin by disentangling network parameters from the input coordinates [99]. Convolutional architectures also achieve this parameter-input decoupling, allowing for larger, more expressive networks that scale effectively to complex, large-scale problems [58].

Some studies have proposed using numerical differentiation (ND)-based methods for the differential operators in the physics-informed loss, with ND being potentially cheaper

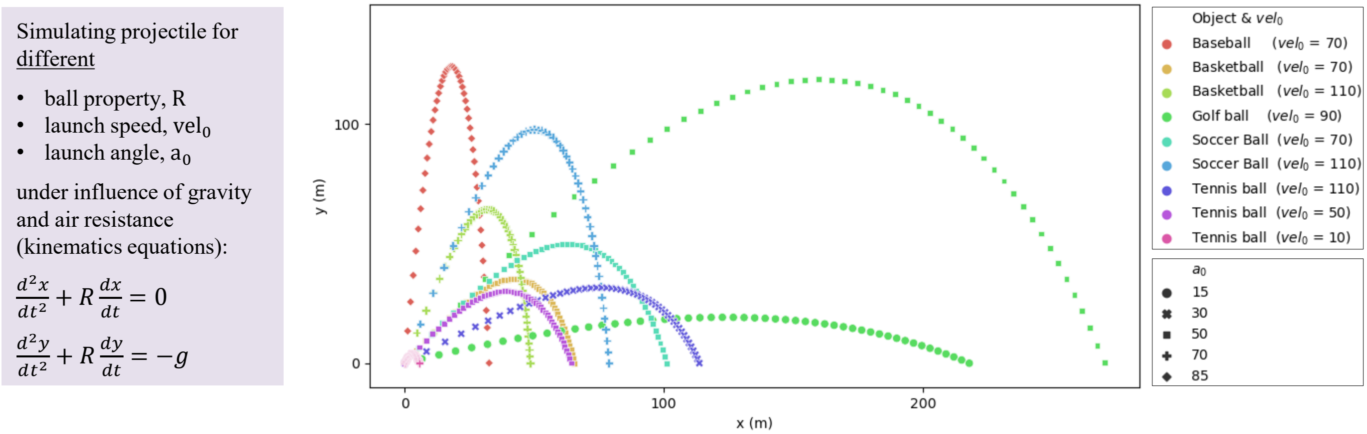


Fig. 6: Example of learning PINN for simulating a distribution of tasks with *a priori* unknown input conditions in a what-if analysis: projectiles of ball with different physics properties and initial conditions (launch angle and launch speed).

to compute relative to automatic differentiation [53, 54]. More importantly, ND is more robust to sample density and potentially advantageous in the context of reducing sample requirement in establishing a strong correlation between physics-informed loss and model accuracy. This is because ND methods, by definition, use Taylor series expansion (TSE) to construct a *low-order local approximation* for the derivative terms. They can provide structural biases and discourage extreme gradients to reduce overfitting to better learn the approximate solution [53]. They can also connect neighboring samples and facilitate the propagation of information from the IC/BC to interior sample points, thereby reducing any tendency to overfit especially near boundaries [54].

B. Challenges in Learning Generalizable PINN Model

Given both simulation (hypothesis evaluation) and inference capabilities, PINNs are versatile and can be applied to various applications common in science such as model-predictive-control, scenario planning, and design exploration [100–103]; see Fig. 6. The commonality to these applications is the need for multiple accurate and physics-compliant predictions across a series of related problems. These repeated evaluations usually differ by modifications to the physics scenario such as changes in geometric shape, PDE parameter, IC, and BC.

1) How Generalizable Is A PINN Model? Most parameterized PINN models (ref. to Subsection II-C; they are designed to learn a parametric family of physics scenarios) make predictions on new scenarios unseen during training through interpolation. This interpolation is usually physics-agnostic, with similar negative implications for generalizability as predictions from typical DNNs [52]. Consequently, the effectiveness of parameterized PINNs depends significantly on problem complexity. Given the difficulties in optimizing a PINN model for a single scenario, the cost of performing physics-informed learning across a set of scenarios can be prohibitive [104]. Moreover, the problem's parameterization needs to be determined at the learning stage, which reduces the flexibility of the parameterized PINN model when applied to new physics scenarios.

2) How to Train A Generalizable PINN Model in The Data-Scarce Regime? Meta-learning approaches provide a systematic framework for discovering good weight initializations that can be quickly adapted—*with minimal training*—to new scenarios (tasks) in a problem distribution. Hence, meta-learning of a generic PINN model initialization can similarly accelerate physically precise predictions across diverse, multi-modal problem distributions, and holds immense promise for addressing generalization challenge of PINN [104–112].

There are still a number of unresolved issues in meta-learning of a generalizable PINN model. Recent PINN studies showed inferior results for model-agnostic meta-learning (MAML), potentially due to insufficient task-specific adaptation and non-convergence in the meta-learning algorithm's outer loop optimization [109, 110]. A fundamental assumption of popular gradient-based meta-learning algorithms—such as MAML [113] and its variant, Reptile [114]—is that the models can learn well on each task after a few gradient descent updates from the initialization. However, this is not the case for PINNs because physics-informed losses are much more challenging to optimize and require extensive training [104]. With only a handful of task-specific gradient descent updates, gradient-based meta-learning algorithms may remain susceptible to local minima from noisy or even deceptive gradients.

IV. Evo-PINN: Physics-Informed Learning with Evolutionary Algorithms

Despite their theoretical advantage in diverse scientific domains, Section III has highlighted several practical challenges in today's PINN learning and optimization. The cost of optimization, navigating spectral bias and local minima traps, has even seeded a sense of pessimism among computational scientists about the utility of these techniques over standard numerical methods [115, 116]. In this section, we present our hypothesis that many algorithmic ideas developed in the computational intelligence community may in fact be uniquely well positioned to tackle the issues plaguing

PINNs. One exciting research direction is to achieve physics-informed learning by synergizing evolutionary algorithms with local gradient descent signals, thereby leveraging the best of both worlds to quickly arrive at physically accurate and generalizable PINN models. In what follows, we examine this and other promising algorithmic directions that we believe to warrant sustained investigation. These directions highlight key opportunities for the evolutionary optimization of PINNs (Evo-PINNs) to advance scientific machine learning.

A. Exploration-Oriented Evolutionary Algorithms for Tackling PINN Optimization Challenges

1) Neuroevolution Can Avoid Local Minima Traps: In looking at a PINN's unique optimization challenges, the need for new algorithms to overcome local minima traps caused by flat outputs at initialization, or by the attractiveness of unstable fixed point solutions to dynamical systems, becomes apparent. Gradient descent is a point-based optimization method that is highly sensitive to the issues of local minima or learning bias magnified by PINN loss functions. In contrast, in silico evolutionary computation offers a class of highly parallelizable approaches for guiding *populations* of candidate solutions towards globally-optimal regions of a search space via randomized mechanisms inspired by biological evolution [117]. We broadly refer to any such algorithmic realization as an *evolutionary algorithm* (EA) and its application to the training of neural networks as *neuroevolution*.

The operators of EAs are typically based on principles of natural selection. They do not require derivative computations of the loss function to be optimized and are therefore gradient-free. Importantly, as population-based methods, the implicit parallelism of EAs implies that they have lesser tendency to get trapped in a single local minimum [49, 118]. What is more, since their search is typically invariant to order-preserving transformations of the loss function [119], EAs are less susceptible to flat output functions with small (vanishing) gradients. For these reasons, they are deemed as potential alternatives to SGD, particularly in the context of scientific machine learning where the goal is to converge to physically precise solutions represented by global optima with zero PINN loss [48]. In addition to its global search capacity, an evolved population naturally provides an ensemble of neural networks that affords the possibility of accurate predictions with robust uncertainty quantification—as was shown through the application of a swarm-based evolutionary procedure to PINN optimization [120, 121]. The ability to provide uncertainty quantification and probabilistic forecasts is crucial for scientific modelling and control problems [122].

2) Pareto Optimization Balances Physics and Data Losses: The existence of multiple loss terms—see Eqs. (2) and (3)—makes PINN training intrinsically a multi-objective optimization problem. Since the terms $\mathcal{L}_{data}, \mathcal{L}_{pde}, \mathcal{L}_{ic}$, and $\mathcal{L}_{bc}$ represent different physical quantities, their magnitudes may be incommensurable. In this case, the role of the scalarizing weights is not only to express preferences between the losses, but also to compensate for differences in their magnitudes. While a perfect physics solution would, in theory, simultaneously satisfy all PINN loss terms, the expressive capacity of a neural network is always limited by its (necessarily) finite size in practice. As argued in [89, 123], capacity limitations naturally give rise to conflicts and trade-offs between loss terms. This, in conjunction with the extreme ruggedness of the physics-informed learning objective, implies that a neural network, despite being a theoretical universal function approximator, is not guaranteed to find a solution that simultaneously minimizes all the terms [46]. When data-driven terms are also included, such as observational data or empirical measurements, true contradictions may arise as the data may be noisy, incomplete, or governed by partially unknown physics that differs from the modeled equations. In such cases, the multi-objective optimization must reconcile inherently conflicting objectives between data fitting and physics adherence. Identifying an appropriate aggregation of objectives to arrive at good (physically satisfactory) trade-off solution is also non-trivial [124] Moreover, the shape of the Pareto front formed by the multiple objectives is not known beforehand [125], with simple weighted-sum aggregations being insufficient to support all parts of non-convex fronts [126].

Multi-objective EAs (MOEAs) are tailor-made to address these issues [127]. In particular, Pareto dominance-based variants of MOEAs [128] do not require the multiple loss terms to be scalarized, and hence are less sensitive to incommensurable objectives. Their population-based search mechanisms also allow for free-form convergence towards the entire Pareto front in a single run of the algorithm, irrespective of the front's shape. These salient features have naturally begun to attract the attention of PINN researchers, although associated developments are still at a nascent stage. In [129], the non-dominated sorting genetic algorithm [130] was hybridized with a variant of SGD (namely, Adam [131]) for training the parameters of a PINN. The exploration capacity of the EA was shown to help overcome local minima and enable precise satisfaction of the initial and boundary conditions encoded in the physics-informed loss. Given the computational cost of a hybrid framework, a pure population-based multi-objective search algorithm was studied for PINN training in [121], leading to calibrated uncertainty quantification by examining the empirical distribution of the ensemble's predictions.

In addition to using MOEAs for PINN parameter training, there have been efforts towards evolving the model's hyperparameters (including the scalarizing weights and activation functions). The method yields a Pareto set of hyperparameters where each setting offers an optimal trade-off between the data and physics losses. Practitioners can then decide post-hoc which setting they wish to pursue [132]. An analysis of such post-hoc decision-making reveals that solutions close to the convex bulge of the Pareto front typically lead to performant PINNs [124]. Evolutionary algorithmists will immediately notice a connection between this observation and the literature on knee-point driven MOEAs [133], which could serve as a valuable resource for hyperparameter tuning in PINNs.

3) Memetic Algorithms Can Navigate Complex PINN Landscapes: There is a clear theoretical rationale supporting Evo-PINNs, yet there are only a handful of related works in the literature. A barrier to the widespread use of EAs has been their high sample complexity, a consequence of updating and evaluating populations of solutions over multiple iterations. It is useful to consider hybridizing global evolutionary search with local gradient signals—thus inheriting the merits of both approaches. While the former can serve as an efficient basin-hopper for exploring rugged PINN landscapes, the latter facilitates rapid convergence to stationary points by exploiting local derivative information of loss functions.

There exists a long history of research in evolutionary computation studying synergies of this kind, with adaptive memetic algorithms forming a class of methods tailored for this purpose [134]. In the deep learning literature, gradient-guided evolutionary approaches have been designed for training neural networks [135, 136], with Cui *et al.* [137] proving a simple theorem that the performance of an evolutionary stochastic gradient descent method with elitist selection strategy never degrades with increasing iterations. However, to the best of our knowledge, there is no comparable body of work for PINN. Davi and Braga-Neto [120] demonstrated that combining a swarm optimization algorithm (a population-based method like EAs) with gradient descent could produce an ensemble of PINN models that out-performed pure gradient-based learning. This result illuminates unexplored avenues for unifying SGD with gradient-free EAs as complementary algorithms within a single framework for finding globally optimum PINNs.

The crafting of synergistic optimization frameworks that meld gradient-based and gradient-free methods, however, remains challenging—particularly in identifying interaction mechanisms that effectively harness the strengths of both approaches. Key questions to address include balancing the intensity of SGD versus EAs, ascertaining when gradient descent should take over from evolution, or selecting an appropriate subset of the evolving population for fine-tuning with gradient descent (as doing so for the entire population would be too expensive). Notably, the subfield of memetic computation—where *memes* are loosely defined as computationally-encoded units of task-specific information (e.g., local gradient signals) combined with global evolutionary search—offers a wealth of theories and methodologies that tackle these issues [138, 139].

4) Implementation Considerations and Hardware Acceleration: While population-based EAs offer search dynamics that complement point-based gradient descent, it is important to recognize that current algorithms may not yet provide an "off-the-shelf" solution for PINN optimization. Indeed, a recent study based on exploratory landscape analysis shows that the features of optimization benchmarks typically used in the development of EAs fundamentally differ from those of neural network training tasks [140]. Excelling at benchmarks does not, therefore, ensure effectiveness in neuroevolution. Although such in-depth analysis in the specific context of PINNs is so far unavailable, an empirical comparison of various evolution strategies (ES) by Sung *et al.* [141] shows that the performance of the classical CMA-ES [142] is statistically similar to or better than modern variants of natural ES algorithms [143] as well as SGD, across a range of PINN tasks representing real-world phenomena in classical mechanics, heat and mass transfer, fluid dynamics, and wave propagation. A compelling reason to consider ES is the induced Gaussian-smoothing of loss functions, biasing flat minima that often depict better generalization performance as per the minimum description length principle [144]. All algorithms tested in [141] were implemented in the JAX framework [145], providing orders of magnitude faster convergence than standard implementations. The source code has been made publicly available[1] to facilitate deeper analysis and development of customized EAs for PINNs.

Not only is there a plethora of EAs to choose from, effective application of an EA to any problem also involves careful tuning of numerous hyperparameters (e.g., learning step-size, population size, crossover rates). Interestingly, the theory of information-geometric optimization (IGO) [119] provides a mathematical unification of probabilistic model-based EAs—including CMA-ES, natural ES, OpenAI-ES [146], or techniques that replace the Gaussian model with a deep generative model [147]—such that even a complete EA can be thought of as just a specific hyperparameter setting of the general IGO update rule. Empirical comparisons of optimizers suggest that more general update rules, if appropriately configured, should not underperform the ones they can approximate [148]. It is therefore fortunate that there exists a valuable body of literature on the automatic hyperparameter configuration of EAs [149] that could be leveraged for PINN optimization problems of varying scales and characteristics. Examples include but are not limited to adaptive population sizing [150] and reinforcement learning for online control of learning step-sizes [151] or crossover rates [149].

Another major challenge in the application of EAs is that they may scale poorly with the dimensionality of the optimized parameters. This number can indeed grow large for deep neural networks, and theoretical analysis suggests that the required number of optimization steps scales linearly with dimensions for general non-smooth functions [152]. However, OpenAI's seminal paper on ES [146] prominently demonstrates that, in practice, these algorithms effectively exploit key informative dimensions of the optimization problem. This allows their performance to scale with the (low) intrinsic dimensionality of the task, rather than with the full dimensionality of the search space. This observation has been supported by recent work as well [5], positioning ES as a scalable option for training large PINNs given the availability of distributed hardware on which population evaluations can be parallelized. For tasks where the intrinsic dimensionality remains high, the latest methods of problem decomposition, cooperative coevolution, and memetic algorithms are worthy of exploration [153], though these have yet to be investigated in the context of PINNs.

[1] Python source codes for neuro-evolving PINN using EvoJAX are available at https://github.com/nicholassung97/Neuroevolution-of-PINNs.

B. Neural Architecture Search for Inducing Physics Biases

The choice of architectures and hyperparameters for PINNs has often been guided by experiential priors drawn from standard deep learning. However, a growing body of literature shows that accurate physics-informed learning may call for bespoke architectures. For example, contrary to popular belief, more hidden layers do not necessarily mean better performance and can sometimes be detrimental. An automated neural architecture search via a bilevel programming formulation revealed that, for certain governing differential equations, a shallow neural network with many neurons was more suitable for PINNs [154]. Similar intriguing insights regarding the choice of activation function have also been uncovered. While squashing functions (e.g., *tanh* and *sigmoid*) and variants of the rectified linear unit are the mainstay in the deep learning literature, the use of Fourier features (i.e., *sine* and *cosine*) has gained prominence in physics tasks [81, 83], specifically for their ability to overcome spectral bias and capture periodic or high frequency patterns (e.g., shock fronts) common in various physics applications [155–157].

Given the large number of design choices for network architectures and hyperparameters, manual experimentation to identify an optimal design for every physical problem is impractical. Notably, there exists a rich history of research in neuroevolution that automates such design choices through the joint evolution of neural network topologies along with their parameters [158, 159]. One such example of the fully automatic design of PINNs by a genetic algorithm with specially tailored selection, crossover, and mutation operators was proposed in [50]. Each chromosome of this method encodes a learning algorithm, architecture of the PINN model, activation function in the hidden layers, and the output activation. In a related approach, an asynchronous genetic algorithm that alternates between model selection and optimization of physics-informed neural state space dynamics models was presented in [160]. Daw *et al.* [87] proposed an evolutionary sampling procedure to incrementally accumulate sample points in regions of high PDE residuals for mitigating training failures.

In order to achieve higher approximation accuracy and faster convergence rates, an EA for concurrently exploring optimal shortcut connections between PINN layers and novel parametric activation functions expressed by binary trees was put forward in [161]. Interestingly, numerical studies show that random projection PINNs—where most network parameters are randomly initialized and fixed, with training limited to the output layer—significantly accelerate training on physics-informed learning objectives [162]. However, ensuring predictive accuracy in this setting requires careful configuration of the randomization bandwidth: a hyperparameter defining the width of the probability distribution from which the random parameters are sampled. In [163], this hyperparameter optimization was carried out by the differential evolution algorithm [164], resulting in PINNs that outperformed higher-order finite element methods (an established numerical method for solving differential equations) in terms of speed and accuracy.

Training a PINN model can be computationally intensive, rendering the evaluation of hyperparameters or network architectures prohibitively expensive. To address this problem, various surrogate-assisted evolutionary techniques [165] have been proposed within the field of evolutionary computing, emerging as promising solutions. These techniques encompass a wide range of optimization scenarios, including single-objective [166], multi-objective [167], large-scale [168], constrained [169], discrete variable [170], and even mixed-variable optimization [171]. These techniques provide a strong foundation for developing more efficient hyperparameter optimization and architecture search methods tailored to PINNs.

C. Transfer Optimization for Accelerated PINN Training

1) Learning from Related Tasks: In a scientific study, it is common for a single set of PDEs to be evaluated under different physics scenarios (tasks), e.g., variations in PDE parameters, and IC/BCs. Moreover, for dynamical problems with causality, PINNs can generally be formulated to predict sequences of physical states, such as at different crack steps in a crack propagation study [172] or at different time steps in a time-varying dynamical problem [173, 174]. Such PINN models are sequentially learned using solutions from previous steps as IC/BCs. Relevant task solving experiences (i.e., optimizing PINN model parameters) can naturally accumulate over time. This allows transfer learning and optimization algorithms to exploit reusable knowledge from past episodes and jump-start the physics-informed learning, hence achieving rapid convergence for new target task [175]. The transfer learning of PINN has also been shown to improve solution accuracy on challenging tasks, such as high-frequency and multi-scale problems, compared to learning the model from scratch [176].

Transfer learning in the context of PINN commonly involves transferring network weights from a similar physics scenario that has been previously learned (source task). These transferred weights serve as initialization for training the target task. Typically, only the last few layers of the networks are optimized after the transfer, allowing the sharing of previously learned useful features between target and source tasks. A singular value decomposition-based transfer approach has also been proposed in [177], which involves freezing the singular vectors of the dense kernel matrix of the hidden layers and only optimizing the singular values for the target task. Apart from network weights, the study from [178] has shown that transfer learning can also facilitate the learning of loss-balancing weights in both forward and inverse PINN models.

Another interesting approach worth exploring is to combine physics-informed learning and data-driven learning techniques using transfer learning. A multi-fidelity transfer learning framework was proposed in [179] to first train a PINN model to capture important information about the governing physics of the problem from the approximated (low-fidelity) physics equations. The model then undergoes data-driven learning with available high-fidelity data, leading to accurate predictions even with limited high-fidelity training data. While the study has demonstrated that transfer learning from physics can benefit a data-driven DNN, the reverse pathway deserves further investigation. There is potential for improvements to PINN optimization by harvesting the

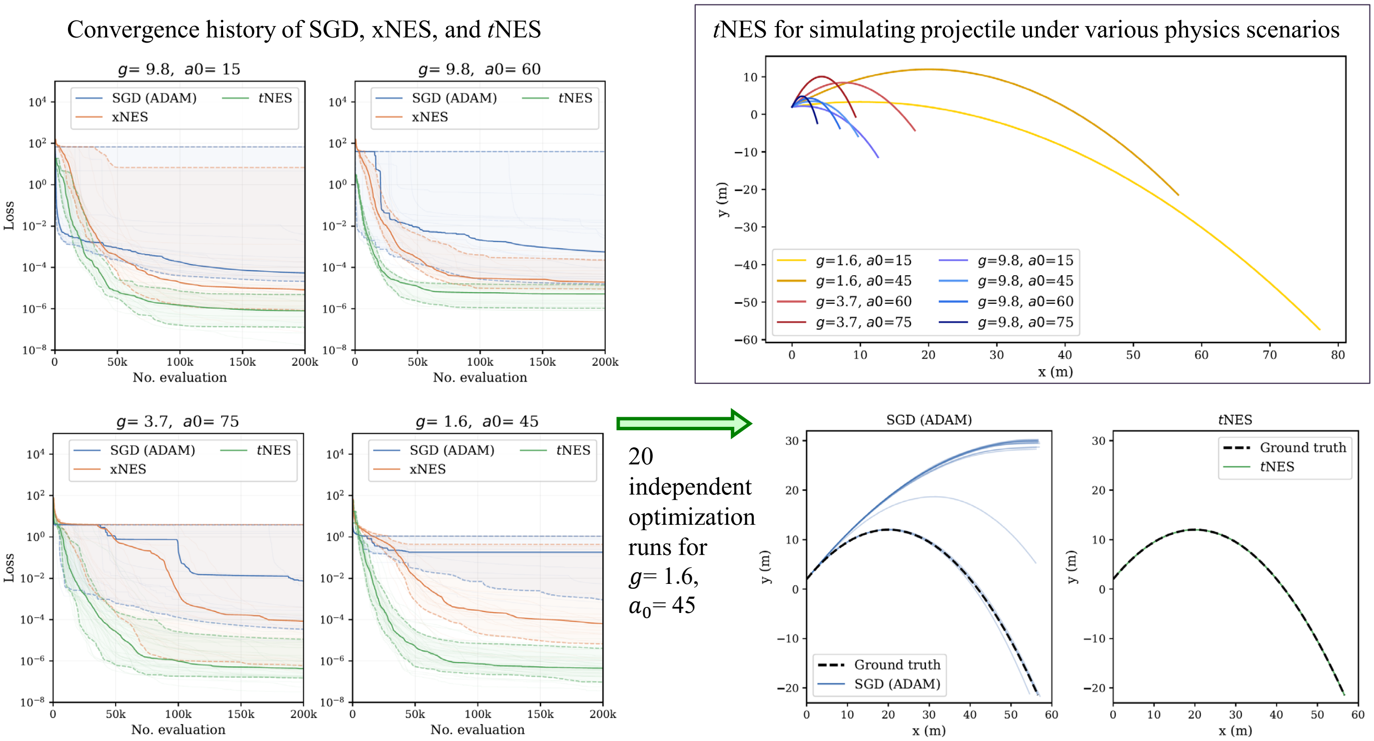


Fig. 7: A PINN simulating projectile motion under gravitational forces on Earth, Mars, or the moon is optimized by transfer natural evolution strategies—dubbed *t*NES. The empirical comparisons of convergence trends between SGD, xNES, and *t*NES for selected scenarios are provided (the bold lines indicate the mean convergence path of multiple independent runs and the shaded areas indicate their 10th-90th interpercentile ranges). Individual solutions given by multiple SGD and *t*NES (transfer from a relevant prior, i.e., source distribution obtained by solving a different gravity g or launch angle a_0) optimization runs for one particular scenario (g=1.6, a_0=45) are plotted and compared against the ground truth. *Adapted with modifications from [48]. Original figure available at [arXiv:2101.01998].*

experience gained from a data-driven DNN model through transfer learning and transfer evolutionary optimization.

2) Adaptive Transfer: An important question in the transfer of physics-informed learning is the identification of relevant source task(s). If the source task is far from the target task, there could be insufficient commonalities between these tasks for the transfer to be beneficial, and it could even negatively impact the learning performance of the target task (negative transfer). For problems with causality, a natural choice for *source* is the networks learned on the previous time-step or physical state [173, 174]. However, the target function is generally *a priori* unknown in physics-informed learning, hence calculating the similarity between target and source tasks based on their output functions is not feasible. Identifying the most relevant source task then becomes a challenge that requires prior knowledge of how the system behaves when its geometry, PDE, and IC/BC parameters change [180]. This becomes even more challenging when dealing with previously solved tasks involving multiple parameter changes.

Adaptive transfer learning can be a particular strategy to address the issues without the need for explicit *a priori* identification of relevant sources. For example, an adaptive transfer physics-informed learning framework was proposed in [180] to achieve effective transfer, even when there is significant variation in the PDE parameters between the source and target tasks. The approach treats the PDE parameters as trainable, hence, they can be adaptively updated from the source to target parameter values, creating a low-loss path in the parameter space to guide the optimization from source to target tasks. Throughout this process, the loss is consistently kept at a low level (similar to the concept of a minimum energy path connecting two minima [181]), ensuring that the PINN solution remains very close to the physical solution and guaranteeing the stability of the training process.

A recent study by Wong *et al.* [48] on the transfer neuroevolution of PINN demonstrated that accurate solutions for new and harder PDEs can be rapidly achieved by optimizing PINN with EA. A key aspect of this transfer optimization approach is its ability to incorporate multiple sources of knowledge while optimizing the PINN for the specific target task. This can be achieved through a principled probability mixture model-based EA that adaptively assigns weights to the different sources based on their observed relevance to the target task during training, thereby preventing detrimental negative transfers. Fig. 7 illustrates the accelerated convergence trends achievable through the transfer neuroevolution of PINN, with clear optimization advantages relative to xNES (a state-of-the-art of EA [143]) and SGD-based (Adam) training.

Similar probabilistic formulations are also possible in multitask neuroevolution [182, 183], enabling the simultaneous solving of sets of related PDEs in a single algorithmic pass. A multitask approach could prove much faster than solving PDEs independently, as useful information is shared between tasks without the need to optimize from scratch.

D. Evolutionary Meta-Learning for Generalizable PINNs

1) Meta-Learning in PINNs: Meta-learning algorithms provide a systematic framework for reusing knowledge from a distribution of learning tasks, allowing models to quickly adapt to new physics-informed learning tasks *with minimal training*. Crucially, meta-learning offers a pathway towards a generalizable PINN over a broad class of physics scenarios, including different PDE families and geometries, as highlighted in Subsection III-B. For example, the MAML and Reptile algorithms have been recently explored in the context of PINN [108–110]. They seek to meta-learn an optimal model initialization in the space of network weights from which the PINN can rapidly converge to a solution that minimizes the physics-informed loss of new tasks. This is achieved by a task-specific fine-tuning procedure during the meta-learning stage through an inner-outer loop training framework. This problem can be mathematically posed as a two-stage or bi-level optimization. The network weights are trained explicitly (in the outer loop) by optimizing the model performance over a batch of physics-informed learning tasks after one or a few gradient descent updates (at the inner loop) from the learned initialization.

2) Neural Baldwinism: Recently, a meta-learning PINN framework based on the evolutionary principle of Baldwinism—dubbed *Baldwinian* PINN [52]—has been proposed to address the training challenges faced in conventional gradient-based meta-learning algorithms such as MAML. The *Baldwin effect* in nature explains the development of phenotypic plasticity among a population's individuals [184], whereby their young are "born ready" with strong learning biases to perform a wide range of tasks. In order to pre-wire such learning ability into the initial connection strengths of a neural network, enabling it to be performant over a distribution of physics-informed learning tasks, an algorithmic realization of neural Baldwinism is examined in the context of PINNs. The algorithm tackles the two-stage optimization problem in meta-learning by coupling hidden layer evolution, for better generalization across tasks, with lifetime-learning at the output layer to specialize to particular tasks. Task-specific learning is limited to the output layer only [185, 186], allowing the model to solve new linear PDE problems with a one-step pseudoinverse operation (or a few pseudoinverse steps for nonlinear PDE problems). The pseudoinverse operation has a closed form and is therefore fast. Unlike gradient-based MAML, the global evolutionary search in Baldwinian-PINNs is better equipped to handle the two-stage/bi-level problem formulation where exact gradients may be difficult to compute. Being a gradient-free algorithm, it does not require unrolling and differentiating through the computation graph, making Baldwinian-PINN an efficient method that scales well with the number of GPUs available. Fig. 8 illustrates the contrast between MAML and Baldwinian evolution for PINN[2].

Experiments in [52] clearly illustrate how Baldwinian-PINN can consistently solve new physics scenarios within a diverse task distribution with orders of magnitude improvement in speed and accuracy compared to a single PINN model. These PDEs include both linear equations such as the convection-diffusion, Helmholtz and Poisson equations, and non-linear equations such as the Burger's, Allen-Cahn and diffusion-reaction equations. Examples of effective generalization of a Baldwinian-PINN to an entire PDE family with diverse output patterns is demonstrated in Fig. 9, on two task scenarios: **S1** shows successful learning of solution for unseen set of PDEs, including changes to PDE parameters, source term, and IC profile; while **S2** shows effective extrapolation of solution to a longer time domain. Baldwinian-PINNs have clear benefits for the continual modeling of dynamical systems through their versatility in handling ICs and ability to rapidly model and stitch time windows together with minimal error. Crucially, Baldwinian evolution does not require explicit parameterization, and evolved Baldwinian-PINN can flexibly generalize to new ICs, BCs, and PDE source terms, a stark contrast to the need for existing parameterized PINNs to have pre-defined PDE, BC or IC parameters. Overall, the outstanding results in [52] compared to other meta-learning methods (Fig. 10) strongly motivate similar approaches in the spirit of Baldwinian-PINN.

3) Meta-Learning Neural Architecture, Loss Function, and Optimizer: Most gradient-based meta-learning PINN approaches proposed in the literature aim to find an optimal initialization of the network weights. In the study by [187], a meta-learning framework for the PINN loss function that incorporates the underlying physics of a parametrized PDE has demonstrated significant performance improvement for new tasks. In [188], the complexity of the PINN optimization problem was proposed to be tackled by *learning* a customized optimizer that's itself parameterized by a neural network. Notably, the learning of the optimizer was done using a persistent ES algorithm [189], and showed that meta-learned optimizers could substantially improve vanilla PINNs over standard hand-crafted optimizers such as Adam.

There are already a handful of studies that have proposed using EAs to find performant neural architectures and hyperparameters for a single PINN (ref. to Subsection IV-B). There is huge scope for future research and development into using EAs to jointly meta-learn model initialization, neural architecture, loss function (hyperparameters), and even the gradient-based optimizer to drastically improve convergence on a set of PINN tasks. For example, an EA can jointly optimize the neural architecture and network weights for a PINN such that the meta-learned PINN model utilizes physics compliance during prediction of a new scenario to enhance generalization across diverse tasks with varying initial conditions, PDE parameters, and geometries.

There is significant potential in exploring evolutionary

[2] Python source codes for Baldwinian-PINN are available at https://github.com/chiuph/Baldwinian-PINN.

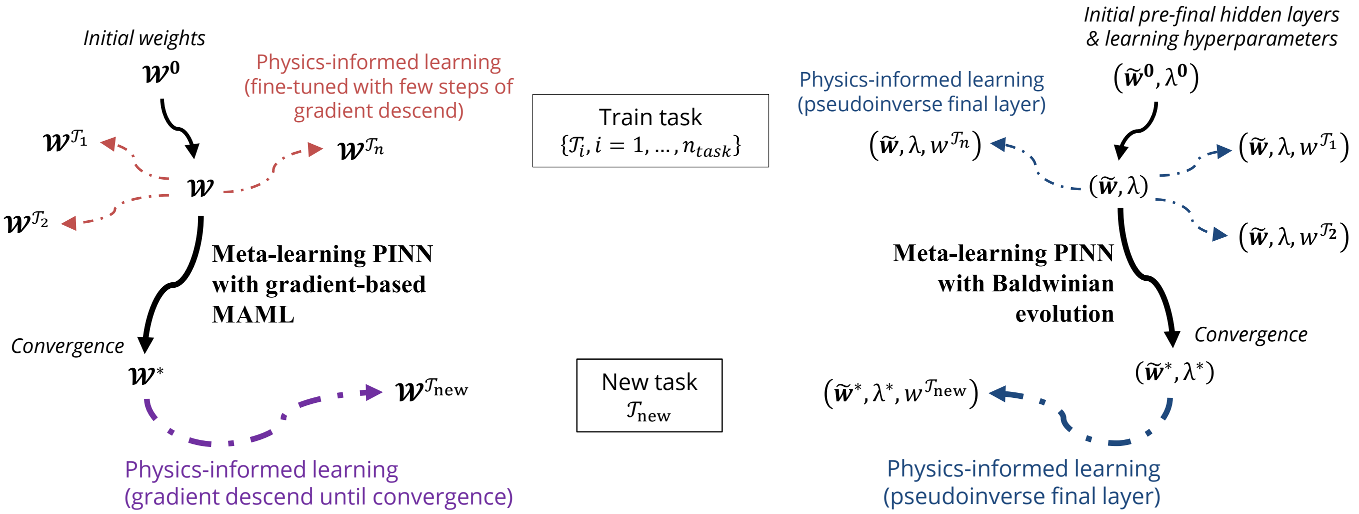

Initial weights
w^0
Physics-informed learning
(fine-tuned with few steps of
gradient descend)
$w^{\mathcal{T}_1}$
$w^{\mathcal{T}_n}$
w
$w^{\mathcal{T}_2}$
Meta-learning PINN
with gradient-based
MAML
Convergence
w^*
$w^{\mathcal{T}_{\text{new}}}$
Physics-informed learning
(gradient descend until convergence)
Train task
$\{\mathcal{T}_i, i = 1, \dots, n_{task}\}$
New task
$\mathcal{T}_{\text{new}}$
Initial pre-final hidden layers
& learning hyperparameters
$(\tilde{w}^0, \lambda^0)$
Physics-informed learning
(pseudoinverse final layer)
$(\tilde{w}, \lambda, w^{\mathcal{T}_n})$
$(\tilde{w}, \lambda)$
$(\tilde{w}, \lambda, w^{\mathcal{T}_1})$
$(\tilde{w}, \lambda, w^{\mathcal{T}_2})$
Meta-learning PINN
with Baldwinian
evolution
Convergence
$(\tilde{w}^*, \lambda^*)$
$(\tilde{w}^*, \lambda^*, w^{\mathcal{T}_{\text{new}}})$
Physics-informed learning
(pseudoinverse final layer)

Fig. 8: Meta-learning PINN with MAML (left) versus Baldwinian evolution (right). In MAML, the initial PINN weights are learned using gradient-based method, such that they can be quickly fine-tuned (physics-informed learning) on new tasks. The task-specific fine-tuning is limited to one or a few gradient descents during training, which is usually insufficient for a meta-learned PINN to converge. In Baldwinian evolution, the weight distribution in the pre-final nonlinear hidden layers $\tilde{w}$ and learning hyperparameters λ are jointly evolved. The task-specific fine-tuning is performed on the output layer weights w (segregated from $\tilde{w}$) with a 1-step pseudoinverse operation at both training and test time (for linear ODE / PDEs). *Adapted with modifications from [52]. Original figure available at [arXiv:2312.03243].*

computation methods to develop meta-learnable optimizers, loss functions, neural architectures, and initializations tailored to diverse physics-informed learning tasks, thereby presenting opportunities to enhance the generalizability of PINN models. Lastly, we would like to make additional remarks on the important role of labeled data in meta-learning PINN. In general, the meta-learning incurs a higher upfront training cost to achieve good generalization. Some proposed methods such as the Baldwinian-PINN cleverly utilize a data-centric meta-learning objective, i.e., using data MSE rather than purely physics-driven loss. Meta-learning PINN algorithms can be meaningfully guided by a few tasks with labeled data, easing optimization difficulties caused by the complex loss landscape defined by the physics-informed loss.

4) Neural Operator Networks: A notable alternative to learning general PDE solutions is the use of neural operator networks *in a purely data-driven setting* [190–192]. By learning mappings between infinite-dimensional function spaces, neural operators enable prediction of solutions in real-time under different PDE source terms, ICs or BCs. For instance, consider the 1D Poisson equation $\frac{d^2u}{dx^2} = f(x)$; a neural operator aims to learn the mapping $G : f \to u$ from the source term $f(x)$ to the PDE solution $u(x)$. Prominent neural operator architectures encompass the Deep Operator Network (DeepONet) [193] and the Fourier Neural Operator (FNO) [194]. DeepONet comprises a branch network to encode the input function at predetermined sample locations, and a trunk network to encode spatial coordinates at which the output function is evaluated. Conversely, FNO employs spectral convolutions to efficiently model nonlocal interactions and directly predict the entire field solution, making it particularly suited for structured-grid data [195]. Further extensions, such as graph-based neural operators [196], transformer operators [197], and geometry-informed operators [198], broaden this framework to manage irregular domains and bolster computational efficiency. Similar to other data-driven methods, neural operators are inherently data-intensive, typically necessitating large PDE simulation datasets to achieve robust generalization.

The integration of physics-informed losses, such as PDE constraints, into neural operators can substantially reduce training data requirements by leveraging known physics [199]. This approach establishes PINNs and neural operators as complementary techniques. Nonetheless, analogous to parameterized PINNs, adherence to physical laws during training does not necessarily translate to their preservation in out-of-sample predictions, thereby presenting common generalization challenges. As a result, neural operators often resort to using larger training datasets to enhance generalization. Furthermore, unlike parameterization-agnostic Baldwinian-PINNs, they demand explicit parameterization of the conditions under which the model is intended to generalize. Although this explicit requirement diminishes the flexibility of neural operators relative to meta-learning strategies, they remain a vital complementary method for generalized neural PDE solvers. Currently, the effectiveness of EAs for optimizing neural operators remains an open area of research. Early work bringing randomization into shallow operator networks has appeared in [200], showing their potential to outperform standard operator networks in both approximation accuracy and computational efficiency.

E. *Evolving Interpretable Models Beyond PINNs*

In scientific machine learning, a key objective of discovering new insights from experimental data relies on human-interpretable models or outputs. While advances have been made in explaining the outputs of deep neural networks, these models are largely perceived as black-boxes [201]. The same

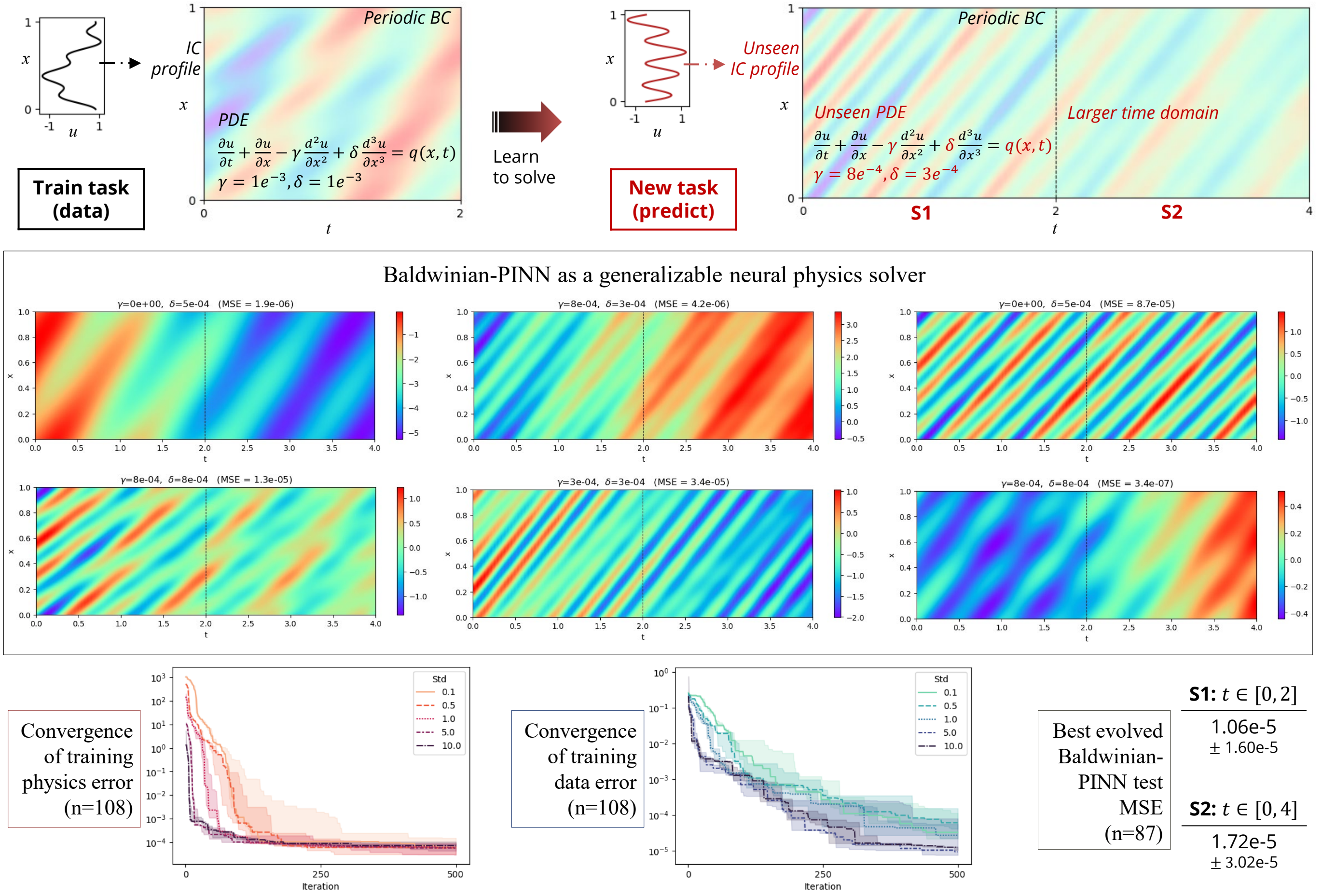


Fig. 9: Schematic to illustrate new tasks arising from *family of PDEs problem*: **S1** change to new PDE (γ and δ parameters), source term q, and IC profile for $t \in [0, 2]$ (same time domain as train tasks), and **S2** projection to longer time domain $t \in [0, 4]$. The solution for new PDE tasks (they are visually indistinguishable from the ground truth) can be quickly obtained in the order of milli-seconds by the Baldwinian-PINN [52]. In [52], CMA-ES was employed to evolve the Baldwinian-PINN models, demonstrating effective convergence in both physics residuals and MSE (108 training tasks). Superior performance is achieved with CMA-ES using initial standard deviations of 5 and 10. The mean MSE across 87 test tasks is below 5e-5.*Adapted with modifications from [52]. Original figure available at [arXiv:2312.03243].*

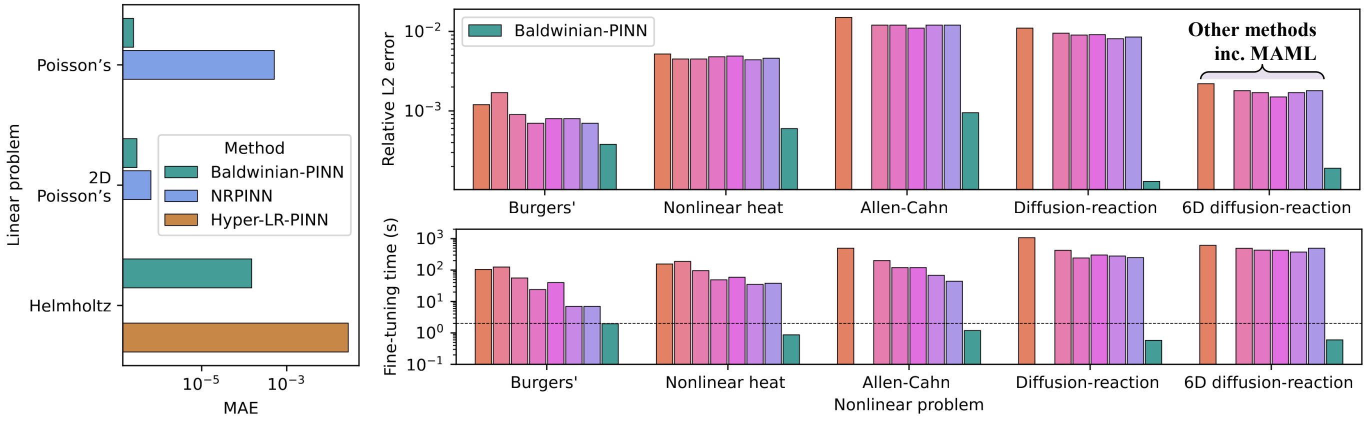


Fig. 10: Empirical comparison between results reported in Baldwinian-PINN [52] and recent meta-learning PINN studies. For linear problems, the results for NRPINN and Hyper-LR-PINN are extracted from [108] and [110], respectively. For nonlinear problems, the results for other meta-learning PINN models / methods (inc. MAML) are extracted from [109]. On nonlinear test tasks, evolved Baldwinian-PINNs achieve up to 10x lower relative norm error, with prediction times under 2 seconds—over 100x faster than other meta-learning methods. For instance, in diffusion-reaction experiments, Baldwinian-PINN yields 70x better accuracy with 700x less computation time.

issue naturally afflicts PINNs, prompting the brief discussion below on the evolution of other model types.

In contrast to neural networks, the subfield of *genetic programming* in evolutionary computation has a long history of successfully discovering free-form natural laws from data through symbolic regression [202, 203]. Recently, these algorithms have even been used for solving high-dimensional PDEs, with the distinct advantage of offering compact analytical expressions approximating complex solutions [204–206]. Balancing compactness and model accuracy is a nontrivial task that is well-suited to evolutionary multi-objective optimization. In addition to symbolic regression, a highly interpretable mathematical description of dynamical systems can be obtained by identifying a best-fit linear dynamics model (i.e., a linear system of governing ODEs) from data. Although finding such a data-driven model that simultaneously satisfies physics constraints translates into a highly constrained, nonlinear and non-convex optimization problem [207], it has been shown to be efficiently solvable by evolution strategies [208]. Adaptations of this evolutionary approach to identify nonlinear dynamics have also been explored [209], with the final prediction accuracy and training stability of evolution exceeding those of models trained by gradient descent.

V. Conclusion

As the influence of deep learning permeates beyond typical natural language processing or computer vision tasks, it is clear that rationalizable models that adhere to the fundamental laws of nature are a need of the hour. Instilling centuries of accumulated wisdom, in the form of mathematically expressible scientific knowledge, into otherwise naive statistical models is crucial. Research efforts in the past few years have demonstrated that PINNs are a promising approach to that end.

However, like any new technique, PINNs inevitably call for new technical innovations. Many of the approaches commonly used in training standard data-driven neural networks need to be re-evaluated, such as the method of network initialization and the choice of activation functions. The use of purely gradient-based optimization, which includes the hugely successful family of SGD algorithms, is especially worthy of reconsideration in the light of analyses summarized in this work. Identified challenges pertaining to local minima traps, fixed point traps, and vanishing gradients of physics-informed learning objectives highlight how these can prove deleterious for PINNs. These obstacles to conventional gradient-based optimization serve to outline ways in which gradient-free algorithms of computational intelligence may be uniquely advantageous as substitutes for or in synergy with SGD. Indeed, the crafting of synergistic optimization frameworks that meld gradient-based and gradient-free optimization unveils novel research questions and directions, firmly positioning scientific machine learning as an emerging area of translational interest for the computational intelligence community.

Acknowledgment

This research was in part supported by the National Research Foundation, Singapore through the AI Singapore Programme, under the project “AI-based urban cooling technology development” (Award No. AISG3-TC-2024-014-SGKR), in part by the National Research Foundation, Singapore and DSO National Laboratories under the AI Singapore Programme - “Design Beyond What You Know: Material Informed Differential Generative AI (MIDGAI) for LightWeight High-Entropy Alloys and Multi-functional Composites (Stage 1b)” (Award No. AISG2-GC-2023-010). Abhishek Gupta is supported by the Ramanujan Fellowship from the Anusandhan National Research Foundation, Government of India (Grant No. RJF/2022/000115).

References

[1] G. Carleo *et al.*, “Machine learning and the physical sciences,” *Reviews of Modern Physics*, vol. 91, no. 4, p. 045002, 2019.

[2] M. Wong, J. Liu, T. Rios, S. Menzel, and Y. S. Ong, “Llm2tea: Agentic ai designer finds innovative objects with generative evolutionary multitasking,” *arXiv preprint arXiv:2406.14917*, 2024.

[3] A. Ghafarollahi and M. J. Buehler, “Automating alloy design and discovery with physics-aware multimodal multiagent ai,” *Proceedings of the National Academy of Sciences*, vol. 122, no. 4, p. e2414074122, 2025.

[4] C. Lu, C. Lu, R. T. Lange, J. Foerster, J. Clune, and D. Ha, “The ai scientist: Towards fully automated open-ended scientific discovery,” *arXiv preprint arXiv:2408.06292*, 2024.

[5] Z. Wei *et al.*, “Evolvable conditional diffusion,” 2025. [Online]. Available: https://arxiv.org/abs/2506.13834

[6] Q. Xu, J. Liu, M. Wong, C. Chen, and Y.-S. Ong, “Looks great, functions better: Physics compliance text-to-3d shape generation.”

[7] Q. Wang, L. Song, Z. Guo, J. Li, and Z. Feng, “A novel multi-fidelity surrogate for efficient turbine design optimization,” *Journal of Turbomachinery*, vol. 146, no. 4, 2024.

[8] J. Zhang, J. Du, M. Zhang, Z. Chen, H. Zhang, and C. Nie, “Aerodynamic performance improvement of a highly loaded compressor airfoil with coanda jet flap,” *Journal of Thermal Science*, vol. 31, pp. 151–162, 2022.

[9] F. Mazé and F. Ahmed, “Diffusion models beat gans on topology optimization,” in *Proceedings of the AAAI conference on artificial intelligence*, vol. 37, no. 8, 2023, pp. 9108–9116.

[10] H.-M. Huang, E. Raponi, F. Duddeck, S. Menzel, and M. Bujny, “Topology optimization of periodic structures for crash and static load cases using the evolutionary level set method,” *Optimization and Engineering*, vol. 25, no. 3, pp. 1597–1630, 2024.

[11] G. Sun, F. Xu, G. Li, and Q. Li, “Crashing analysis and multiobjective optimization for thin-walled structures with functionally graded thickness,” *International Journal of Impact Engineering*, vol. 64, pp. 62–74, 2014.

[12] Y. Xiong *et al.*, “Data-driven design space exploration and exploitation for design for additive manufacturing,” *Journal of Mechanical Design*, vol. 141, no. 10, p. 101101, 2019.

[13] J. Doh, N. Raju, N. Raghavan, D. W. Rosen, and S. 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*, vol. 155, p. 106535, 2022.

[14] X. Lai, S. Wang, Z. Guo, C. Zhang, W. Sun, and X. Song, “Designing a shape–performance integrated digital twin based on multiple models and dynamic data: a boom crane example,” *Journal of Mechanical Design*, vol. 143, no. 7, p. 071703, 2021.

[15] X. He, L. Yang, Z. Gong, Y. Pang, J. Li, Z. Kan, and X. Song, “Digital twin-based online structural optimization? yes, it’s possible!” *Thin-Walled Structures*, p. 112796, 2024.

[16] B.-C. Wang, Z.-D. Ji, Y. Wang, H.-X. Li, and Z. Li, “A physics-informed composite network for modeling of electrochemical

process of large-scale lithium-ion batteries," *IEEE Transactions on Industrial Informatics*, 2024.

[17] Y. Zou, Y. Zeng, S. Li, and Q. Zhu, "Machine learning model with physical constraints for diffuse optical tomography," *Biomedical optics express*, vol. 12, no. 9, pp. 5720–5735, 2021.

[18] X. Garbet, Y. Idomura, L. Villard, and T. Watanabe, "Gyrokinetic simulations of turbulent transport," *Nuclear Fusion*, vol. 50, no. 4, p. 043002, 2010.

[19] X. Garbet *et al.*, "The effect of shaping on trapped electron mode stability: an analytical model," *Nuclear Fusion*, vol. 64, no. 10, p. 106055, Sep. 2024.

[20] S. A. Niaki, E. Haghighat, T. Campbell, A. Poursartip, and R. Vaziri, "Physics-informed neural network for modelling the thermochemical curing process of composite-tool systems during manufacture," *Computer Methods in Applied Mechanics and Engineering*, vol. 384, p. 113959, 2021.

[21] R. Roscher, B. Bohn, M. F. Duarte, and J. Garcke, "Explainable machine learning for scientific insights and discoveries," *Ieee Access*, vol. 8, pp. 42 200–42 216, 2020.

[22] D. Gunning and D. Aha, "Darpa's explainable artificial intelligence (xai) program," *AI magazine*, vol. 40, no. 2, pp. 44–58, 2019.

[23] Y.-S. Ong and A. Gupta, "Air 5: Five pillars of artificial intelligence research," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 3, no. 5, pp. 411–415, 2019.

[24] A. Karpatne *et al.*, "Theory-guided data science: A new paradigm for scientific discovery from data," *IEEE Transactions on knowledge and data engineering*, vol. 29, no. 10, pp. 2318–2331, 2017.

[25] V. Rueden *et al.*, "Informed machine learning–a taxonomy and survey of integrating prior knowledge into learning systems," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 1, pp. 614–633, 2021.

[26] J. Willard, X. Jia, S. Xu, M. Steinbach, and V. Kumar, "Integrating scientific knowledge with machine learning for engineering and environmental systems," *ACM Computing Surveys*, vol. 55, no. 4, pp. 1–37, 2022.

[27] G. E. Karniadakis, I. G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, and L. Yang, "Physics-informed machine learning," *Nature Reviews Physics*, vol. 3, no. 6, pp. 422–440, 2021.

[28] T. Cohen and M. Welling, "Group equivariant convolutional networks," in *International conference on machine learning*. PMLR, 2016, pp. 2990–2999.

[29] G. Barbastathis, A. Ozcan, and G. Situ, "On the use of deep learning for computational imaging," *Optica*, vol. 6, no. 8, pp. 921–943, 2019.

[30] P. W. Battaglia *et al.*, "Relational inductive biases, deep learning, and graph networks," *arXiv preprint arXiv:1806.01261*, 2018.

[31] M. Raissi, P. Perdikaris, and G. 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*, vol. 378, pp. 686–707, 2019.

[32] H. Lee and I. S. Kang, "Neural algorithm for solving differential equations," *Journal of Computational Physics*, vol. 91, no. 1, pp. 110–131, 1990.

[33] I. E. Lagaris, A. Likas, and D. I. Fotiadis, "Artificial neural networks for solving ordinary and partial differential equations," *IEEE transactions on neural networks*, vol. 9, no. 5, pp. 987–1000, 1998.

[34] J. Bradbury *et al.* (2018) JAX: composable transformations of Python+NumPy programs. [Online]. Available: http://github.com/jax-ml/jax

[35] Nvidia physicsnemo: An open-source framework for physics-based deep learning in science and engineeringg. [Online]. Available: https://github.com/NVIDIA/physicsnemo

[36] E. Haghighat and R. 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*, vol. 373, p. 113552, 2021.

[37] L. Lu, X. Meng, Z. Mao, and G. E. Karniadakis, "Deepxde: A deep learning library for solving differential equations," *SIAM review*, vol. 63, no. 1, pp. 208–228, 2021.

[38] A. Krishnapriyan, A. Gholami, S. Zhe, R. Kirby, and M. W. Mahoney, "Characterizing possible failure modes in physics-informed neural networks," *Advances in Neural Information Processing Systems*, vol. 34, pp. 26 548–26 560, 2021.

[39] O. Fuks and H. A. Tchelepi, "Limitations of physics informed machine learning for nonlinear two-phase transport in porous media," *Journal of Machine Learning for Modeling and Computing*, vol. 1, no. 1, 2020.

[40] W. Ji, W. Qiu, Z. Shi, S. Pan, and S. Deng, "Stiff-pinn: Physics-informed neural network for stiff chemical kinetics," *The Journal of Physical Chemistry A*, vol. 125, no. 36, pp. 8098–8106, 2021.

[41] S. Cuomo, V. S. Di Cola, F. Giampaolo, G. Rozza, M. Raissi, and F. Piccialli, "Scientific machine learning through physics–informed neural networks: Where we are and what's next," *Journal of Scientific Computing*, vol. 92, no. 3, p. 88, 2022.

[42] S. Cai, Z. Mao, Z. Wang, M. Yin, and G. E. Karniadakis, "Physics-informed neural networks (pinns) for fluid mechanics: A review," *Acta Mechanica Sinica*, vol. 37, no. 12, pp. 1727–1738, 2021.

[43] C. Banerjee, K. Nguyen, C. Fookes, and K. George, "Physics-informed computer vision: A review and perspectives," *ACM Computing Surveys*, vol. 57, no. 1, pp. 1–38, 2024.

[44] Z. Hao, J. Yao, C. Su, H. Su, Z. Wang, F. Lu, Z. Xia, Y. Zhang, S. Liu, L. Lu *et al.*, "Pinnacle: A comprehensive benchmark of physics-informed neural networks for solving pdes," *arXiv preprint arXiv:2306.08827*, 2023.

[45] J. D. Toscano, V. Oommen, A. J. Varghese, Z. Zou, N. Ahmadi Daryakenari, C. Wu, and G. E. Karniadakis, "From pinns to pikans: Recent advances in physics-informed machine learning," *Machine Learning for Computational Science and Engineering*, vol. 1, no. 1, pp. 1–43, 2025.

[46] R. Bischof and M. Kraus, "Multi-objective loss balancing for physics-informed deep learning," *arXiv preprint arXiv:2110.09813*, 2021.

[47] S. Wang, Y. Teng, and P. Perdikaris, "Understanding and mitigating gradient flow pathologies in physics-informed neural networks," *SIAM Journal on Scientific Computing*, vol. 43, no. 5, pp. A3055–A3081, 2021.

[48] J. C. Wong, A. Gupta, and Y.-S. Ong, "Can transfer neuroevolution tractably solve your differential equations?" *IEEE Computational Intelligence Magazine*, vol. 16, no. 2, pp. 14–30, 2021.

[49] A. Gupta and B. Mishra, "Neuroevolving monotonic pinns for particle breakage analysis," in *2024 IEEE Conference on Artificial Intelligence (CAI)*. IEEE, 2024, pp. 993–996.

[50] A. Kaplarević-Mališić, B. Andrijević, F. Bojović, S. Nikolić, L. Krstić, B. Stojanović, and M. Ivanović, "Identifying optimal architectures of physics-informed neural networks by evolutionary strategy," *Applied Soft Computing*, vol. 146, p. 110646, 2023.

[51] A. Daw, J. Bu, S. Wang, P. Perdikaris, and A. Karpatne, "Mitigating propagation failures in physics-informed neural networks using retain-resample-release (r3) sampling," in *Proceedings of the 40th International Conference on Machine Learning*, 2023, pp. 7264–7302.

[52] J. C. Wong *et al.*, "The baldwin effect in advancing generalizability of physics-informed neural networks," *arXiv preprint arXiv:2312.03243*, 2024.

[53] P.-H. Chiu, J. C. Wong, C. Ooi, M. H. Dao, and Y.-S. Ong, "Can-pinn: A fast physics-informed neural network based on coupled-automatic–numerical differentiation method," *Com-*

puter Methods in Applied Mechanics and Engineering, vol. 395, p. 114909, 2022.

[54] J. C. Wong, P.-H. Chiu, C. Ooi, M. H. Dao, and Y.-S. Ong, "Lsa-pinn: Linear boundary connectivity loss for solving pdes on complex geometry," in *2023 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2023, pp. 1–10.

[55] M. Stynes, "Steady-state convection-diffusion problems," *Acta Numerica*, vol. 14, pp. 445–508, 2005.

[56] R. Temam, *Navier–Stokes equations: theory and numerical analysis*. American Mathematical Society, 2024, vol. 343.

[57] J. Peiró and S. Sherwin, "Finite difference, finite element and finite volume methods for partial differential equations," *Handbook of Materials Modeling: Methods*, pp. 2415–2446, 2005.

[58] N. Wandel, M. Weinmann, and R. Klein, "Learning incompressible fluid dynamics from scratch–towards fast, differentiable fluid models that generalize," *arXiv preprint arXiv:2006.08762*, 2020.

[59] R. Zhang, Y. Liu, and H. Sun, "Physics-informed multi-lstm networks for metamodeling of nonlinear structures," *Computer Methods in Applied Mechanics and Engineering*, vol. 369, p. 113226, 2020.

[60] M. Mahmoudabadbozchelou and S. Jamali, "Rheology-informed neural networks (rhinns) for forward and inverse metamodelling of complex fluids," *Scientific reports*, vol. 11, no. 1, pp. 1–13, 2021.

[61] L. Yang, X. Meng, and G. E. Karniadakis, "B-pinns: Bayesian physics-informed neural networks for forward and inverse pde problems with noisy data," *Journal of Computational Physics*, vol. 425, p. 109913, 2021.

[62] F. A. Viana and A. K. Subramaniyan, "A survey of bayesian calibration and physics-informed neural networks in scientific modeling," *Archives of Computational Methods in Engineering*, vol. 28, pp. 3801–3830, 2021.

[63] A. M. Tartakovsky, C. O. Marrero, P. Perdikaris, G. D. Tartakovsky, and D. Barajas-Solano, "Physics-informed deep neural networks for learning parameters and constitutive relationships in subsurface flow problems," *Water Resources Research*, vol. 56, no. 5, p. e2019WR026731, 2020.

[64] Z. Chen, Y. Liu, and H. Sun, "Physics-informed learning of governing equations from scarce data," *Nature communications*, vol. 12, no. 1, p. 6136, 2021.

[65] E. Kharazmi, M. Cai, X. Zheng, Z. Zhang, G. Lin, and G. E. Karniadakis, "Identifiability and predictability of integer- and fractional-order epidemiological models using physics-informed neural networks," *Nature Computational Science*, vol. 1, no. 11, pp. 744–753, 2021.

[66] R. Shi, Z. Mo, and X. 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*, vol. 35, no. 1, 2021, pp. 540–547.

[67] C. Oszkinat, S. E. Luczak, and I. Rosen, "Uncertainty quantification in estimating blood alcohol concentration from transdermal alcohol level with physics-informed neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

[68] M. Raissi, A. Yazdani, and G. E. Karniadakis, "Hidden fluid mechanics: Learning velocity and pressure fields from flow visualizations," *Science*, vol. 367, no. 6481, pp. 1026–1030, 2020.

[69] Q. Zhu, Z. Liu, and J. Yan, "Machine learning for metal additive manufacturing: predicting temperature and melt pool fluid dynamics using physics-informed neural networks," *Computational Mechanics*, vol. 67, pp. 619–635, 2021.

[70] X. Liu, W. Peng, Z. Gong, W. Zhou, and W. Yao, "Temperature field inversion of heat-source systems via physics-informed neural networks," *Engineering Applications of Artificial Intelligence*, vol. 113, p. 104902, 2022.

[71] S. Cai, H. Li, F. Zheng, F. Kong, M. Dao, G. E. Karniadakis, and S. Suresh, "Artificial intelligence velocimetry and microaneurysm-on-a-chip for three-dimensional analysis of blood flow in physiology and disease," *Proceedings of the National Academy of Sciences*, vol. 118, no. 13, p. e2100697118, 2021.

[72] C. Ruiz Herrera, T. Grandits, G. Plank, P. Perdikaris, F. Sahli Costabal, and S. Pezzuto, "Physics-informed neural networks to learn cardiac fiber orientation from multiple electroanatomical maps," *Engineering with Computers*, vol. 38, no. 5, pp. 3957–3973, 2022.

[73] L. Lu, R. Pestourie, W. Yao, Z. Wang, F. Verdugo, and S. G. Johnson, "Physics-informed neural networks with hard constraints for inverse design," *SIAM Journal on Scientific Computing*, vol. 43, no. 6, pp. B1105–B1132, 2021.

[74] Z. Fang and J. Zhan, "Deep physical informed neural networks for metamaterial design," *IEEE Access*, vol. 8, pp. 24 506–24 513, 2019.

[75] S. Wang, H. Wang, and P. Perdikaris, "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*, vol. 384, p. 113938, 2021.

[76] S. Wang, X. Yu, and P. Perdikaris, "When and why pinns fail to train: A neural tangent kernel perspective," *Journal of Computational Physics*, vol. 449, p. 110768, 2022.

[77] M. Elhamod, J. Bu, C. Singh, M. Redell, A. Ghosh, V. Podolskiy, W.-C. Lee, and A. Karpatne, "Cophy-pgnn: Learning physics-guided neural networks with competing loss functions for solving eigenvalue problems," *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2020.

[78] Z. Xiang, W. Peng, X. Liu, and W. Yao, "Self-adaptive loss balanced physics-informed neural networks," *Neurocomputing*, vol. 496, pp. 11–34, 2022.

[79] C. Wang, S. Li, D. He, and L. Wang, "Is $\mathcal{L}^2$ physics-informed loss always suitable for training physics-informed neural network?" *arXiv preprint arXiv:2206.02016*, 2022.

[80] S. Wang, S. Sankaran, and P. Perdikaris, "Respecting causality for training physics-informed neural networks," *Computer Methods in Applied Mechanics and Engineering*, vol. 421, p. 116813, 2024.

[81] J. C. Wong, C. C. Ooi, A. Gupta, and Y.-S. Ong, "Learning in sinusoidal spaces with physics-informed neural networks," *IEEE Transactions on Artificial Intelligence*, vol. 5, no. 3, pp. 985–1000, 2022.

[82] A. Gupta and B. K. Mishra, "Multi-head neural networks for simulating particle breakage dynamics," *Theoretical and Applied Mechanics Letters*, vol. 14, no. 2, p. 100515, 2024.

[83] G. Jin, J. C. Wong, A. Gupta, S. Li, and Y.-S. Ong, "Fourier warm start for physics-informed neural networks," *Engineering Applications of Artificial Intelligence*, vol. 132, p. 107887, 2024.

[84] N. Rahaman *et al.*, "On the spectral bias of neural networks," in *International Conference on Machine Learning*. PMLR, 2019, pp. 5301–5310.

[85] Z.-Q. J. Xu, Y. Zhang, and Y. Xiao, "Training behavior of deep neural network in frequency domain," in *Neural Information Processing: 26th International Conference, ICONIP 2019, Sydney, NSW, Australia, December 12–15, 2019, Proceedings, Part I 26*. Springer, 2019, pp. 264–274.

[86] A. Bonfanti, G. Bruno, and C. Cipriani, "The challenges of the nonlinear regime for physics-informed neural networks," *arXiv preprint arXiv:2402.03864*, 2024.

[87] A. Daw, J. Bu, S. Wang, P. Perdikaris, and A. Karpatne, "Rethinking the importance of sampling in physics-informed neural networks," *arXiv preprint arXiv:2207.02338*, 2022.

[88] S. Wang and P. Perdikaris, "Long-time integration of parametric evolution equations with physics-informed deeponets," *Journal of Computational Physics*, vol. 475, p. 111855, 2023.

[89] F. M. Rohrhofer, S. Posch, C. Gößnitzer, and B. Geiger, "On the role of fixed points of dynamical systems in training physics-informed neural networks," *Transactions on Machine Learning Research*, vol. 2023, no. 1, p. 490, 2023.

[90] G. K. R. Lau, A. Hemachandra, S.-K. Ng, and B. K. H. Low, "Pinnacle: Pinn adaptive collocation and experimental points selection," *arXiv preprint arXiv:2404.07662*, 2024.

[91] C. C. Ooi *et al.*, "Importance of nyquist-shannon sampling in training of physics-informed neural networks," in *2024 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2024, pp. 1–8.

[92] H. Wu, H. Luo, Y. Ma, J. Wang, and M. Long, "Ropinn: Region optimized physics-informed neural networks," *arXiv preprint arXiv:2405.14369*, 2024.

[93] S. Wang, S. Sankaran, H. Wang, and P. Perdikaris, "An expert's guide to training physics-informed neural networks," *arXiv preprint arXiv:2308.08468*, 2023.

[94] S. Wang, B. Li, Y. Chen, and P. Perdikaris, "Piratenets: Physics-informed deep learning with residual adaptive networks," *arXiv preprint arXiv:2402.00326*, 2024.

[95] S. Wang, A. K. Bhartari, B. Li, and P. Perdikaris, "Gradient alignment in physics-informed neural networks: A second-order optimization perspective," *arXiv preprint arXiv:2502.00604*, 2025.

[96] V. Sitzmann, J. Martel, A. Bergman, D. Lindell, and G. Wetzstein, "Implicit neural representations with periodic activation functions," *Advances in Neural Information Processing Systems*, vol. 33, pp. 7462–7473, 2020.

[97] A. D. Jagtap, K. Kawaguchi, and G. E. Karniadakis, "Adaptive activation functions accelerate convergence in deep and physics-informed neural networks," *Journal of Computational Physics*, vol. 404, p. 109136, 2020.

[98] S. Zeng, Z. Zhang, and Q. Zou, "Adaptive deep neural networks methods for high-dimensional partial differential equations," *Journal of Computational Physics*, vol. 463, p. 111232, 2022.

[99] N. Kang, B. Lee, Y. Hong, S.-B. Yun, and E. Park, "Pixel: Physics-informed cell representations for fast and accurate pde solvers," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, no. 7, 2023, pp. 8186–8194.

[100] F. Arnold and R. King, "State–space modeling for control based on physics-informed neural networks," *Engineering Applications of Artificial Intelligence*, vol. 101, p. 104195, 2021.

[101] G. Gokhale, B. Claessens, and C. Develder, "Physics informed neural networks for control oriented thermal modeling of buildings," *Applied Energy*, vol. 314, p. 118852, 2022.

[102] U. Güngördü and J. Kestner, "Robust quantum gates using smooth pulses and physics-informed neural networks," *Physical Review Research*, vol. 4, no. 2, p. 023155, 2022.

[103] S. Mowlavi and S. Nabi, "Optimal control of pdes using physics-informed neural networks," *Journal of Computational Physics*, vol. 473, p. 111731, 2023.

[104] T. Iwata, Y. Tanaka, and N. Ueda, "Meta-learning of physics-informed neural networks for efficiently solving newly given pdes," *arXiv preprint arXiv:2310.13270*, 2023.

[105] F. de Avila Belbute-Peres, Y.-f. Chen, and F. Sha, "Hyperpinn: Learning parameterized differential equations with physics-informed hypernetworks," in *The symbiosis of deep learning and differential equations*, 2021.

[106] X. Huang *et al.*, "Meta-auto-decoder for solving parametric partial differential equations," *Advances in Neural Information Processing Systems*, vol. 35, pp. 23 426–23 438, 2022.

[107] M. Toloubidokhti, Y. Ye, R. Missel, X. Jiang, N. Kumar, R. Shrestha, and L. Wang, "Dats: Difficulty-aware task sampler for meta-learning physics-informed neural networks," in *The Twelfth International Conference on Learning Representations*, 2023.

[108] X. Liu, X. Zhang, W. Peng, W. Zhou, and W. Yao, "A novel meta-learning initialization method for physics-informed neural networks," *Neural Computing and Applications*, vol. 34, no. 17, pp. 14 511–14 534, 2022.

[109] M. Penwarden, S. Zhe, A. Narayan, and R. M. Kirby, "A metalearning approach for physics-informed neural networks (pinns): Application to parameterized pdes," *Journal of Computational Physics*, vol. 477, p. 111912, 2023.

[110] W. Cho, K. Lee, D. Rim, and N. Park, "Hypernetwork-based meta-learning for low-rank physics-informed neural networks," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[111] Y. Chen and S. Koohy, "Gpt-pinn: Generative pre-trained physics-informed neural networks toward non-intrusive meta-learning of parametric pdes," *Finite Elements in Analysis and Design*, vol. 228, p. 104047, 2024.

[112] W. Cho *et al.*, "Parameterized physics-informed neural networks for parameterized pdes," *arXiv preprint arXiv:2408.09446*, 2024.

[113] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *International conference on machine learning*. PMLR, 2017, pp. 1126–1135.

[114] A. Nichol and J. Schulman, "Reptile: a scalable metalearning algorithm," *arXiv preprint arXiv:1803.02999*, vol. 2, no. 3, p. 4, 2018.

[115] N. McGreivy and A. Hakim, "Weak baselines and reporting biases lead to overoptimism in machine learning for fluid-related partial differential equations," *Nature Machine Intelligence*, vol. 6, no. 10, pp. 1256–1269, 2024.

[116] T. G. Grossmann, U. J. Komorowska, J. Latz, and C.-B. Schönlieb, "Can physics-informed neural networks beat the finite element method?" *IMA Journal of Applied Mathematics*, vol. 89, no. 1, pp. 143–174, 2024.

[117] R. Miikkulainen and S. Forrest, "A biological perspective on evolutionary computation," *Nature Machine Intelligence*, vol. 3, no. 1, pp. 9–15, 2021.

[118] T. Back, *Evolutionary algorithms in theory and practice: evolution strategies, evolutionary programming, genetic algorithms*. Oxford university press, 1996.

[119] Y. Ollivier, L. Arnold, A. Auger, and N. Hansen, "Information-geometric optimization algorithms: A unifying picture via invariance principles," *Journal of Machine Learning Research*, vol. 18, no. 18, pp. 1–65, 2017.

[120] C. Davi and U. Braga-Neto, "Pso-pinn: physics-informed neural networks trained with particle swarm optimization," *arXiv preprint arXiv:2202.01943*, 2022.

[121] ——, "Multi-objective pso-pinn," in *1st Workshop on the Synergy of Scientific and Machine Learning Modeling@ ICML2023*, 2023.

[122] U. Fasel, J. N. Kutz, B. W. Brunton, and S. L. Brunton, "Ensemble-sindy: Robust sparse model discovery in the low-data, high-noise limit, with active learning and control," *Proceedings of the Royal Society A*, vol. 478, no. 2260, p. 20210904, 2022.

[123] H. X. Choong, Y.-S. Ong, A. Gupta, C. Chen, and R. Lim, "Jack and masters of all trades: one-pass learning sets of model sets from large pre-trained models," *IEEE Computational Intelligence Magazine*, vol. 18, no. 3, pp. 29–40, 2023.

[124] Z. Wei *et al.*, "How to select physics-informed neural networks in the absence of ground truth: a pareto front-based strategy," in *1st Workshop on the Synergy of Scientific and Machine Learning Modeling@ ICML2023*, 2023.

[125] F. M. Rohrhofer, S. Posch, C. Gößnitzer, and B. C. Geiger, "On the apparent pareto front of physics-informed neural networks," *IEEE Access*, 2023.

[126] F. Heldmann, S. Berkhahn, M. Ehrhardt, and K. Klamroth, "Pinn training using biobjective optimization: The trade-off between data loss and residual loss," *Journal of Computational Physics*, vol. 488, p. 112211, 2023.

[127] C. A. C. Coello, *Evolutionary algorithms for solving multi-*

objective problems. Springer, 2007.

[128] K. Deb and M. Ehrgott, "On generalized dominance structures for multi-objective optimization," *Mathematical and Computational Applications*, vol. 28, no. 5, p. 100, 2023.

[129] B. Lu, C. Moya, and G. Lin, "Nsga-pinn: a multi-objective optimization method for physics-informed neural network training," *Algorithms*, vol. 16, no. 4, p. 194, 2023.

[130] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: Nsga-ii," *IEEE transactions on evolutionary computation*, vol. 6, no. 2, pp. 182–197, 2002.

[131] D. P. Kingma, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[132] T. de Wolff, H. C. Lincopi, L. Martí, and N. Sanchez-Pi, "Mopinns: an evolutionary multi-objective approach to physics-informed neural networks," in *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, 2022, pp. 228–231.

[133] G. Yu, L. Ma, Y. Jin, W. Du, Q. Liu, and H. Zhang, "A survey on knee-oriented multiobjective evolutionary optimization," *IEEE transactions on evolutionary computation*, vol. 26, no. 6, pp. 1452–1472, 2022.

[134] Y.-S. Ong, M.-H. Lim, N. Zhu, and K.-W. Wong, "Classification of adaptive memetic algorithms: a comparative study," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 36, no. 1, pp. 141–152, 2006.

[135] S. Yang, Y. Tian, C. He, X. Zhang, K. C. Tan, and Y. Jin, "A gradient-guided evolutionary approach to training deep neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 9, pp. 4861–4875, 2021.

[136] Y. Xue, Y. Tong, and F. Neri, "An ensemble of differential evolution and adam for training feed-forward neural networks," *Information Sciences*, vol. 608, pp. 453–471, 2022.

[137] X. Cui, W. Zhang, Z. Tüske, and M. Picheny, "Evolutionary stochastic gradient descent for optimization of deep neural networks," *Advances in neural information processing systems*, vol. 31, 2018.

[138] X. Chen, Y.-S. Ong, M.-H. Lim, and K. C. Tan, "A multi-facet survey on memetic computation," *IEEE Transactions on evolutionary computation*, vol. 15, no. 5, pp. 591–607, 2011.

[139] A. Gupta and Y.-S. Ong, *Memetic computation: the mainspring of knowledge transfer in a data-driven optimization era*. Springer, 2018, vol. 21.

[140] K. M. Malan and M. A. Muñoz, "Why we should be benchmarking evolutionary algorithms on neural network training tasks," in *Proceedings of the Genetic and Evolutionary Computation Conference*, 2025, pp. 30–38.

[141] N. Sung, J. C. Wong, C. C. Ooi, A. Gupta, P.-H. Chiu, and Y.-S. Ong, "Neuroevolution of physics-informed neural nets: benchmark problems and comparative results," in *Proceedings of the Companion Conference on Genetic and Evolutionary Computation*, 2023, pp. 2144–2151.

[142] N. Hansen and A. Ostermeier, "Completely derandomized self-adaptation in evolution strategies," *Evolutionary computation*, vol. 9, no. 2, pp. 159–195, 2001.

[143] D. Wierstra, T. Schaul, T. Glasmachers, Y. Sun, J. Peters, and J. Schmidhuber, "Natural evolution strategies," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 949–980, 2014.

[144] S. Hochreiter and J. Schmidhuber, "Flat minima," *Neural computation*, vol. 9, no. 1, pp. 1–42, 1997.

[145] Y. Tang, Y. Tian, and D. Ha, "Evojax: Hardware-accelerated neuroevolution," in *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, 2022, pp. 308–311.

[146] T. Salimans, J. Ho, X. Chen, S. Sidor, and I. Sutskever, "Evolution strategies as a scalable alternative to reinforcement learning," *arXiv preprint arXiv:1703.03864*, 2017.

[147] B. Hartl, Y. Zhang, H. Hazan, and M. Levin, "Heuristically adaptive diffusion-model evolutionary strategy," *arXiv preprint arXiv:2411.13420*, 2024.

[148] D. Choi, C. J. Shallue, Z. Nado, J. Lee, C. J. Maddison, and G. E. Dahl, "On empirical comparisons of optimizers for deep learning," *arXiv preprint arXiv:1910.05446*, 2019.

[149] M. Tessari and G. Iacca, "Reinforcement learning based adaptive metaheuristics," in *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, 2022, pp. 1854–1861.

[150] K. Nishida and Y. Akimoto, "Psa-cma-es: Cma-es with population size adaptation," in *Proceedings of the Genetic and Evolutionary Computation Conference*, 2018, pp. 865–872.

[151] M. Nomura, Y. Akimoto, and I. Ono, "Cma-es with learning rate adaptation," *ACM Transactions on Evolutionary Learning*, vol. 5, no. 1, pp. 1–28, 2025.

[152] Y. Nesterov and V. Spokoiny, "Random gradient-free minimization of convex functions," *Foundations of Computational Mathematics*, vol. 17, no. 2, pp. 527–566, 2017.

[153] M. N. Omidvar, X. Li, and X. Yao, "A review of population-based metaheuristics for large-scale black-box global optimization—part i," *IEEE Transactions on Evolutionary Computation*, vol. 26, no. 5, pp. 802–822, 2021.

[154] Y. Wang and L. Zhong, "Nas-pinn: neural architecture search-guided physics-informed neural network for solving pdes," *Journal of Computational Physics*, vol. 496, p. 112603, 2024.

[155] T. Zhang, R. Yan, S. Zhang, D. Yang, and A. Chen, "Application of fourier feature physics-information neural network in model of pipeline conveying fluid," *Thin-Walled Structures*, vol. 198, p. 111693, 2024.

[156] O. Sallam and M. Fürth, "On the use of fourier features-physics informed neural networks (ff-pinn) for forward and inverse fluid mechanics problems," *Proceedings of the Institution of Mechanical Engineers, Part M: Journal of Engineering for the Maritime Environment*, vol. 237, no. 4, pp. 846–866, 2023.

[157] Q. A. Huhn, M. E. Tano, and J. C. Ragusa, "Physics-informed neural network with fourier features for radiation transport in heterogeneous media," *Nuclear Science and Engineering*, vol. 197, no. 9, pp. 2484–2497, 2023.

[158] K. O. Stanley, J. Clune, J. Lehman, and R. Miikkulainen, "Designing neural networks through neuroevolution," *Nature Machine Intelligence*, vol. 1, no. 1, pp. 24–35, 2019.

[159] Y. Liu, Y. Sun, B. Xue, M. Zhang, G. G. Yen, and K. C. Tan, "A survey on evolutionary neural architecture search," *IEEE transactions on neural networks and learning systems*, vol. 34, no. 2, pp. 550–570, 2021.

[160] E. Skomski, J. Drgoňa, and A. Tuor, "Automating discovery of physics-informed neural state space models via learning and evolution," in *Learning for Dynamics and Control*. PMLR, 2021, pp. 980–991.

[161] B. Zhang and C. Yang, "Discovering physics-informed neural networks model for solving partial differential equations through evolutionary computation," *Swarm and Evolutionary Computation*, vol. 88, p. 101589, 2024.

[162] S. Dong and Z. Li, "Local extreme learning machines and domain decomposition for solving linear and nonlinear partial differential equations," *Computer Methods in Applied Mechanics and Engineering*, vol. 387, p. 114129, 2021.

[163] S. Dong and J. 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*, vol. 463, p. 111290, 2022.

[164] R. Storn and K. Price, "Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces," *Journal of global optimization*, vol. 11, pp. 341–359, 1997.

[165] Y. Jin, H. Wang, T. Chugh, D. Guo, and K. Miettinen, "Data-driven evolutionary optimization: An overview and case studies," *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 3, pp. 442–458, 2018.

[166] H. Wang, Y. Jin, and J. Doherty, "Committee-based active learning for surrogate-assisted particle swarm optimization of expensive problems," *IEEE transactions on cybernetics*, vol. 47, no. 9, pp. 2664–2677, 2017.

[167] Z. Wang, Q. Zhang, Y.-S. Ong, S. Yao, H. Liu, and J. Luo, "Choose appropriate subproblems for collaborative modeling in expensive multiobjective optimization," *IEEE Transactions on Cybernetics*, vol. 53, no. 1, pp. 483–496, 2021.

[168] C. Sun, Y. Jin, R. Cheng, J. Ding, and J. Zeng, "Surrogate-assisted cooperative swarm optimization of high-dimensional expensive problems," *IEEE Transactions on Evolutionary Computation*, vol. 21, no. 4, pp. 644–660, 2017.

[169] Y. Wang, D.-Q. Yin, S. Yang, and G. Sun, "Global and local surrogate-assisted differential evolution for expensive constrained optimization problems with inequality constraints," *IEEE transactions on cybernetics*, vol. 49, no. 5, pp. 1642–1656, 2018.

[170] H. Wang and Y. Jin, "A random forest-assisted evolutionary algorithm for data-driven constrained multiobjective combinatorial optimization of trauma systems," *IEEE transactions on cybernetics*, vol. 50, no. 2, pp. 536–549, 2018.

[171] J. Liu, Y. Wang, G. Sun, and T. Pang, "Multisurrogate-assisted ant colony optimization for expensive optimization problems with continuous and categorical variables," *IEEE Transactions on Cybernetics*, vol. 52, no. 11, pp. 11 348–11 361, 2021.

[172] S. Goswami, C. Anitescu, S. Chakraborty, and T. Rabczuk, "Transfer learning enhanced physics informed neural network for phase-field modeling of fracture," *Theoretical and Applied Fracture Mechanics*, vol. 106, p. 102447, 2020.

[173] R. Mattey and S. 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*, vol. 390, p. 114474, 2022.

[174] H. Tang, Y. Liao, H. Yang, and L. Xie, "A transfer learning-physics informed neural network (tl-pinn) for vortex-induced vibration," *Ocean Engineering*, vol. 266, p. 113101, 2022.

[175] K. Prantikos, S. Chatzidakis, L. H. Tsoukalas, and A. Heifetz, "Physics-informed neural network with transfer learning (tl-pinn) based on domain similarity measure for prediction of nuclear reactor transients," *Scientific Reports*, vol. 13, no. 1, p. 16840, 2023.

[176] A. H. Mustajab, H. Lyu, Z. Rizvi, and F. Wuttke, "Physics-informed neural networks for high-frequency and multi-scale problems using transfer learning," *Applied Sciences*, vol. 14, no. 8, p. 3204, 2024.

[177] Y. Gao, K. C. Cheung, and M. K. Ng, "Svd-pinns: Transfer learning of physics-informed neural networks via singular value decomposition," in *2022 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE, 2022, pp. 1443–1450.

[178] C. Xu, B. T. Cao, Y. Yuan, and G. Meschke, "Transfer learning based physics-informed neural networks for solving inverse problems in engineering structures under different loading scenarios," *Computer Methods in Applied Mechanics and Engineering*, vol. 405, p. 115852, 2023.

[179] S. Chakraborty, "Transfer learning based multi-fidelity physics informed deep neural network," *Journal of Computational Physics*, vol. 426, p. 109942, 2021.

[180] Y. Liu, W. Liu, X. Yan, S. Guo, and C.-a. Zhang, "Adaptive transfer learning for pinn," *Journal of Computational Physics*, vol. 490, p. 112291, 2023.

[181] T. Garipov, P. Izmailov, D. Podoprikhin, D. P. Vetrov, and A. G. Wilson, "Loss surfaces, mode connectivity, and fast ensembling of dnns," *Advances in neural information processing systems*, vol. 31, 2018.

[182] A. Gupta, L. Zhou, Y.-S. Ong, Z. Chen, and Y. Hou, "Half a dozen real-world applications of evolutionary multitasking, and more," *IEEE Computational Intelligence Magazine*, vol. 17, no. 2, pp. 49–66, 2022.

[183] N. Zhang, A. Gupta, Z. Chen, and Y.-S. Ong, "Multitask neuroevolution for reinforcement learning with long and short episodes," *IEEE Transactions on Cognitive and Developmental Systems*, 2022.

[184] K. L. Downing, "Heterochronous neural baldwinism," in *Artificial Life Conference Proceedings*. Citeseer, 2012, pp. 37–44.

[185] Z. Zou and G. E. Karniadakis, "L-hydra: Multi-head physics-informed neural networks," *arXiv preprint arXiv:2301.02152*, 2023.

[186] S. Desai, M. Mattheakis, H. Joy, P. Protopapas, and S. Roberts, "One-shot transfer learning of physics-informed neural networks," *arXiv preprint arXiv:2110.11286*, 2021.

[187] A. F. Psaros, K. Kawaguchi, and G. E. Karniadakis, "Meta-learning pinn loss functions," *Journal of computational physics*, vol. 458, p. 111121, 2022.

[188] A. Bihlo, "Improving physics-informed neural networks with meta-learned optimization," *Journal of Machine Learning Research*, vol. 25, no. 14, pp. 1–26, 2024.

[189] P. Vicol, L. Metz, and J. Sohl-Dickstein, "Unbiased gradient estimation in unrolled computation graphs with persistent evolution strategies," in *International Conference on Machine Learning*. PMLR, 2021, pp. 10 553–10 563.

[190] S. Goswami, A. Bora, Y. Yu, and G. E. Karniadakis, "Physics-informed deep neural operator networks," in *Machine learning in modeling and simulation: methods and applications*. Springer, 2023, pp. 219–254.

[191] K. Azizzadenesheli, N. Kovachki, Z. Li, M. Liu-Schiaffini, J. Kossaifi, and A. Anandkumar, "Neural operators for accelerating scientific simulations and design," *Nature Reviews Physics*, vol. 6, no. 5, pp. 320–328, 2024.

[192] L. Cao, H. Hong, and M. Jiang, "Fast solving partial differential equations via imitative fourier neural operator," in *2024 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2024, pp. 1–8.

[193] L. Lu, P. Jin, G. Pang, Z. Zhang, and G. E. Karniadakis, "Learning nonlinear operators via deeponet based on the universal approximation theorem of operators," *Nature machine intelligence*, vol. 3, no. 3, pp. 218–229, 2021.

[194] Z. Li *et al.*, "Physics-informed neural operator for learning partial differential equations," *ACM/JMS Journal of Data Science*, vol. 1, no. 3, pp. 1–27, 2024.

[195] Z. Li, D. Z. Huang, B. Liu, and A. Anandkumar, "Fourier neural operator with learned deformations for pdes on general geometries," *Journal of Machine Learning Research*, vol. 24, no. 388, pp. 1–26, 2023.

[196] N. Kovachki, Z. Li, B. Liu, K. Azizzadenesheli, K. Bhattacharya, A. Stuart, and A. Anandkumar, "Neural operator: Learning maps between function spaces with applications to pdes," *Journal of Machine Learning Research*, vol. 24, no. 89, pp. 1–97, 2023.

[197] Z. Hao *et al.*, "Gnot: A general neural operator transformer for operator learning," in *International Conference on Machine Learning*. PMLR, 2023, pp. 12 556–12 569.

[198] Z. Li, N. Kovachki, C. Choy, B. Li, J. Kossaifi, S. Otta, M. A. Nabian, M. Stadler, C. Hundt, K. Azizzadenesheli *et al.*, "Geometry-informed neural operator for large-scale 3d pdes," *Advances in Neural Information Processing Systems*, vol. 36, pp. 35 836–35 854, 2023.

[199] S. Wang, H. Wang, and P. Perdikaris, "Learning the solution operator of parametric partial differential equations with physics-informed deeponets," *Science advances*, vol. 7, no. 40, p. eabi8605, 2021.

[200] G. Fabiani, I. G. Kevrekidis, C. Siettos, and A. N. Yannacopoulos, "Randonets: Shallow networks with random projections for learning linear and nonlinear operators," *Journal of Computational Physics*, vol. 520, p. 113433, 2025.

[201] S. Ali, T. Abuhmed, S. El-Sappagh, K. Muhammad, J. M. Alonso-Moral, R. Confalonieri, R. Guidotti, J. Del Ser, N. Díaz-Rodríguez, and F. Herrera, "Explainable artificial

intelligence (xai): What we know and what is left to attain trustworthy artificial intelligence," *Information fusion*, vol. 99, p. 101805, 2023.

[202] M. Schmidt and H. Lipson, "Distilling free-form natural laws from experimental data," *science*, vol. 324, no. 5923, pp. 81–85, 2009.

[203] J. Dong and J. Zhong, "Recent advances in symbolic regression," *ACM Computing Surveys*, vol. 57, no. 11, pp. 1–37, 2025.

[204] L. Cao *et al.*, "An interpretable approach to the solutions of high-dimensional partial differential equations," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 18, 2024, pp. 20 640–20 648.

[205] L. Cao, Z. Zheng, C. Ding, J. Cai, and M. Jiang, "Genetic programming symbolic regression with simplification-pruning operator for solving differential equations," in *International Conference on Neural Information Processing*. Springer, 2023, pp. 287–298.

[206] L. Cao, Z. Lin, K. C. Tan, and M. Jiang, "Interpretable solutions for multi-physics pdes using t-nngp," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 39, no. 13, 2025, pp. 14 212–14 220.

[207] T. Askham and J. N. Kutz, "Variable projection methods for an optimized dynamic mode decomposition," *SIAM Journal on Applied Dynamical Systems*, vol. 17, no. 1, pp. 380–416, 2018.

[208] A. Gupta and B. K. Mishra, "Globally optimized dynamic mode decomposition: A first study in particulate systems modelling," *Theoretical and Applied Mechanics Letters*, p. 100563, 2024.

[209] J. Hazelden, Y. H. Liu, E. Shlizerman, and E. Shea-Brown, "Evolutionary algorithms as an alternative to backpropagation for supervised training of biophysical neural networks and neural odes," *arXiv preprint arXiv:2311.10869*, 2023.