Advances in Differential Equations and Control Processes 2026, 33(3), 4514.
doi: https://doi.org/10.59400/adecp4514

Review

Frontier advances in solving fractional differential equations via physics-informed neural networks: A comprehensive review

Yi Tian[1], **Xiuqing Peng**[2], **Ji-Huan He**[3,*]

[1] College of Intelligent Science and Technology, Inner Mongolia University of Technology, Hohhot 010080, China
[2] School of Economics and Management, Inner Mongolia University, Hohhot 010021, China
[3] School of Information Engineering, Yango University, Fuzhou 350015, China
*** Corresponding authors:** Ji-Huan He, hejihuan@ygu.edu.cn

CITATION

Tian Y, Peng X, He JH. Frontier advances in solving fractional differential equations via physics-informed neural networks: A comprehensive review. Advances in Differential Equations and Control Processes. 2026; 33(3): 4514. https://doi.org/10.59400/adecp4514

ARTICLE INFO

Received: 16 June 2026
Revised: 15 July 2026
Accepted: 20 July 2026
Available online: 28 July 2026

Abstract: Fractional differential equations (FDEs) serve as indispensable mathematical instruments for modeling complex dynamical systems characterized by inherent memory retention, hereditary evolution, and anomalous diffusion—behaviors that elude accurate description by conventional integer-order differential models. Traditional numerical algorithms for FDEs are plagued by substantial mesh discretization overhead, prohibitive computational costs, and poor compatibility with various inverse parameter identification tasks. As a groundbreaking computational framework, physics-informed neural networks (PINNs) synergistically combine data-driven fitting capabilities with rigorous physical governing laws, offering transformative solutions to general differential equations. Cutting-edge advancements in fractional PINNs (fPINNs) are systematically synthesized in this comprehensive review. Physical motivations for adopting FDEs are elaborated, the fundamental architecture of vanilla PINNs is delineated, and core obstacles originating from the non-local integral characteristics of fractional derivative operators are dissected to reveal critical restrictions on direct application of standard PINNs to FDEs. All mainstream fPINN approaches are taxonomically grouped into three distinct technical branches, with focused mathematical derivations and in-depth discussions dedicated to fractional-order inverse identification. A homotopy-augmented simultaneous training scheme is formulated to mitigate unstable joint optimization of fractional orders and neural network weights, enabling a smooth transition of optimization targets from simplistic integer-order dynamics toward complex fractional non-local responses. Relative advantages and disadvantages of simultaneous and two-stage sequential identification workflows are thoroughly contrasted, and clear, actionable criteria are delivered to facilitate selection of appropriate identification pipelines for diverse fractional modeling scenarios. Systematic theoretical references and operable technical guidance for fractional modeling and scientific machine learning are provided throughout the review.

Keywords: fractal-fractional differential equations; physics-informed neural networks; fractional order identification; memory effects; inverse problems

1. Introduction

Fractional differential equations (FDEs) have become essential mathematical tools for describing complex dynamical systems that cannot be accurately represented by traditional integer-order models. By generalizing differentiation and integration to arbitrary real orders, FDEs inherently incorporate memory effects, hereditary properties, and non-local behaviors that are omnipresent in physical, biological, and engineering systems [1–3]. These distinctive properties have made fractional calculus widely applicable across diverse research fields, such as heat and mass transfer in anomalous media [4, 5], industrial equipment modeling and control [6], thermo-viscoelastic analysis of functional materials [7], thermal response analysis of solid structures under complex thermal loads [8], and vibration behavior in porous media systems modeled with fractal-fractional approaches [9]. The growing recognition that many real-world processes—from viscoelastic deformation and anomalous diffusion to electrochemical reactions and biological population dynamics—exhibit memory-dependent dynamics has motivated sustained research efforts in both theoretical development and numerical methods for FDEs [10,11].

Over the past two decades, substantial progress has been achieved in theoretical derivation, numerical algorithm design, and practical applications of FDEs. Traditional analytical methods, including the homotopy perturbation method [12, 13], the variational iteration method [14,15], the exp-function method [16–18], and fixed-point theorems [19], have established a solid theoretical foundation for solving FDEs. However, these conventional approaches exhibit notable limitations: they are often computationally expensive for high-dimensional and strongly nonlinear FDEs; they rely on complete observational data and explicit problem formulations; and most critically, they adapt poorly to inverse problems and parameter identification tasks where governing equations, fractional orders, or system parameters are partially unknown. Furthermore, traditional numerical solvers require mesh generation and careful discretization, which become prohibitively burdensome for problems with complex geometries or irregular domains.

With the rapid development of artificial intelligence and machine learning [20–23], physics-informed neural networks (PINNs) have emerged as a revolutionary paradigm for solving differential equations [24]. Introduced by Raissi et al. originally in two arXiv preprints released in 2017 [25, 26], and then in their well-known 2019 journal article [27], PINNs embed physical governing equations, initial conditions, and boundary constraints directly into the loss function of a neural network, thereby combining the power of data-driven learning with the interpretability of physical law constraints [24,27].

PINNs stand out as a powerful paradigm owing to their inherent strengths: they deliver robust performance under sparse or noisy measurement data [28,29], eliminate the heavy overhead of mesh-based numerical discretization [30,31], and accommodate both forward solution and inverse parameter identification tasks [32]. Subsequent methodological extensions have enriched the original framework with diverse advanced variants, including domain decomposition architectures (XPINN, hp-VPINN) [33,34], Bayesian uncertainty quantification frameworks (B-PINN) [35], transfer learning pipelines [36], adaptive collocation resampling strategies [37], variational and energy-consistent formulations [38], as well as graph- and operator-aware neural designs [39,40]—all devised to boost training convergence, computational scalability, and predictive accuracy. The mesh-free nature of PINNs, coupled with automatic differentiation for accurate derivative computation, makes them particularly attractive for problems where traditional solvers struggle

Recent research has placed central emphasis on three dominant developmental trajectories: customized network architectures [41], advanced optimization schemes [42], and hybrid workflows integrating deep learning with classical numerical solvers, e.g., homotopy-preserving physics-informed neural network [43].

Distinct from conventional purely data-driven neural networks, PINNs enable PDE solving without abundant labeled training samples and generally require far fewer observational data points. This versatile framework supports both state prediction and unknown parameter calibration across a broad spectrum of physical engineering systems [44–46]. Its practical scope has continuously expanded to cover fluid dynamics, heat and mass transfer, solid mechanics, electromagnetics, geophysical modeling, material science, and fractional-order control systems. Collectively, the aforementioned variants have substantially ameliorated longstanding pain points, including unstable training, slow convergence, insufficient enforcement of boundary constraints, and inadequate uncertainty quantification. As a result, PINNs have evolved into an indispensable mainstream tool for scientific computing targeting partial differential equations and their associated inverse problems.

In recent years, extending PINNs to solve FDEs has become an active interdisciplinary research hotspot [47–49]. The central challenge lies in the fundamental incompatibility between the nonlocal nature of fractional operators and the local, chain-rule-based automatic differentiation that underpins standard PINNs. While standard PINNs compute integer-order derivatives exactly via automatic differentiation, fractional derivatives involve convolution integrals over the entire solution history, requiring numerical discretization, specialized loss function design, and often significantly increased computational cost. Despite these obstacles, researchers have developed diverse strategies—including fractional-order modified PINNs, network-structure-enhanced architectures, and hybrid numerical-PINN approaches—that have demonstrated promising results across a range of fractional differential equations and application domains.

This review provides a comprehensive and systematic synthesis of recent advances in solving FDEs via PINNs. Unlike existing surveys that focus primarily on methodological taxonomies or specific application areas, our review places particular emphasis on: (i) a thorough exposition of the mathematical foundations of various fractional derivative definitions and their implications for PINN-based solution strategies; (ii) a critical comparison of existing fractional PINN variants with practical selection guidelines; (iii) a rigorous mathematical formulation for fractional order identification, with detailed comparison of simultaneous versus two-stage sequential strategies; and (iv) an in-depth discussion of open challenges, theoretical gaps, and promising future directions. We aim to equip researchers with both the conceptual understanding and practical guidance necessary to apply fractional PINNs effectively and to identify key opportunities for further innovation.

The remainder of this review is organized as follows. Section 2 elaborates on the rationale for employing FDEs in science and engineering, surveying the major fractional derivative definitions—including Riemann–Liouville, Grünwald–Letnikov, Caputo, Riesz, Caputo–Fabrizio, Atangana–Baleanu, conformable, and two-scale fractal derivatives—and their respective mathematical properties and computational implications. Section 3 introduces the fundamentals of PINNs and explains why standard PINNs cannot be directly applied to FDEs, identifying three fundamental obstacles: non-locality incompatibility, approximation necessity, and computational overhead. Section 4 categorizes existing fractional PINN variants into three mainstream branches, comparing their respective strengths and limitations. Section 5 presents a comprehensive mathematical formulation for fractional-order identification, analyzing both simultaneous and two-stage strategies with extensions to multi-order and variable-order systems. Section 6 concludes the review with key achievements, persistent challenges, and an outlook on future research directions.

2. Rationale for using fractional differential equations

While classical integer-order differential equations perform well in integer-dimensional spaces, many real-world physical, biological, and industrial systems exhibit memory effects, hereditary behavior, and non-local dynamic properties that cannot be adequately captured by integer-order models. Typical examples span self-similar fractal spaces across all scales [50] and two-scale fractal spaces, where non-self-similar porous media serve as a representative instance [51]. These limitations have motivated the development of fractional calculus, which generalizes integer-order differentiation and integration to arbitrary real orders, providing a natural mathematical framework for modeling systems with long-range temporal or spatial correlations.

Frontier progress in solving fractional differential equations with PINNs is moving toward hybrid, adaptive, and operator-aware architectures that improve accuracy, convergence, and scalability beyond standard PINNs. The strongest themes are: fractional-specific discretization, adaptive loss/sampling, domain decomposition/singularity handling, Monte Carlo and quadrature acceleration for high dimensions, and transform/functional-connection reformulations that avoid direct fractional autodiff issues [52–54].

2.1. Major fractional derivative definitions

A variety of fractional derivative definitions have been proposed in the literature, each with distinct mathematical properties, computational characteristics, and suitability for different application contexts. This subsection surveys the most widely adopted definitions, providing a foundation for understanding their integration with PINN frameworks in subsequent sections.

Caputo fractional derivative [55]. For a function $f(t)$, the Caputo fractional derivative of order α ($0 < \alpha < 1$) is defined as:

$$D_t^{\alpha} f(t) = \frac{1}{\Gamma(1-\alpha)} \int_0^t \frac{f'(\tau)}{(t-\tau)^{\alpha}} d\tau. \tag{1}$$

This formulation inherently involves a weighted integral over the entire history from 0 to t, with the kernel $(t - \tau)^{-\alpha}$ assigning greater weight to recent states while still retaining long-term memory. The Caputo derivative has become the most widely adopted definition for physical modeling due to three decisive advantages: (i) it yields zero derivatives for constant functions; (ii) it accommodates physically meaningful initial conditions expressed in terms of integer-order derivatives (e.g., initial displacement and velocity); and (iii) it aligns naturally with conventional integer-order initial value formulations. For viscoelastic materials, fluid flows, and electrochemical systems, Diethelm et al. [55] demonstrated that FDEs using Caputo derivatives capture hereditary properties with fewer parameters and higher accuracy compared to integer-order models. Caputo derivatives have become a standard tool for memory-based fractional modeling across multiple application domains [56,57].

Riemann–Liouville derivative [58]. The Riemann–Liouville (RL) definition proceeds by first applying a fractional-order integral of order $n-\alpha$ (where $n=\lceil\alpha\rceil$) and then taking an integer-order derivative of order n:

$$^{\mathrm{RL}}D^{\alpha}_{a+}f(t)=\frac{1}{\Gamma(n-\alpha)}\left(\frac{d}{dt}\right)^{n}\int_{a}^{t}\frac{f(\tau)}{(t-\tau)^{\alpha-n+1}}\,d\tau. \tag{2}$$

The RL derivative possesses strong mathematical foundations and is widely used in theoretical analysis. However, it suffers from a critical limitation: the fractional derivative of a constant is not zero, but rather $^{\mathrm{RL}}D^{\alpha}_{a+}C=C(t-a)^{-\alpha}/\Gamma(1-\alpha)$ [58,59]. This property has profound consequences for physical modeling, as initial conditions must be specified in terms of fractional integrals that lack a clear physical interpretation. Moreover, the RL derivative exhibits a "super-singular" kernel that complicates its application in engineering contexts [60]. While the RL and Caputo derivatives are equivalent for sufficiently smooth functions with homogeneous initial conditions, the Caputo formulation is almost universally preferred for initial value problems in applied sciences.

Grünwald–Letnikov derivative [61]. The Grünwald–Letnikov (GL) definition adopts a fundamentally different approach, formulated through the limit of a finite difference series:

$$^{\mathrm{GL}}D^{\alpha}_{t}f(t)=\lim_{h\to 0}h^{-\alpha}\sum_{k=0}^{\lfloor (t-a)/h\rfloor}(-1)^{k}\binom{\alpha}{k}f(t-kh). \tag{3}$$

The GL derivative is naturally suited for numerical implementation, as it directly yields discrete approximation schemes for fractional derivatives. For smooth functions, the GL derivative is equivalent to the RL derivative. However, the GL formulation is less frequently employed for theoretical analysis due to the complexity of its series representation. Its primary value lies in computational settings, where the finite-difference interpretation facilitates straightforward discretization—a property that has made it particularly attractive for fractional PINN implementations, as discussed in Section 4.

Riesz fractional derivative [62]. For problems involving spatial nonlocality—such as anomalous diffusion in heterogeneous media, turbulence modeling, and quantum mechanics—the Riesz fractional derivative provides a symmetric, Fourier-based definition particularly suited to problems on unbounded domains. The Riesz derivative is defined as a fractional power of the negative Laplacian:

$$
{}^{\mathrm{RZ}}D_x^{\alpha} f(x) = \Gamma(1+\alpha) \frac{\sin(\pi\alpha/2)}{\pi} \int_0^{\infty} \frac{f(x+\xi) - 2f(x) + f(x-\xi)}{\xi^{\alpha+1}} \, d\xi, 0 < \alpha < 1. \tag{4}
$$

The Riesz operator is inherently nonlocal, coupling the function value at a point with its values throughout the entire spatial domain via the symmetric kernel $| \ \xi \ |^{-(\alpha+1)}$. This symmetry is essential for maintaining Hermiticity in quantum mechanical applications, such as the fractional Schrödinger equation. In the context of fractional diffusion equations, the Riesz derivative naturally produces symmetric, Lévy-flight-type spreading that cannot be captured by one-sided fractional derivatives. The computational implementation of Riesz derivatives poses challenges beyond those of temporal fractional derivatives: the integral extends over the entire spatial domain, leading to $\mathcal{O}(N^2)$ complexity for naive implementations, and the symmetric kernel demands careful treatment of boundary conditions. For problems on finite domains, specialized formulations or spectral methods are typically required.

Caputo–Fabrizio derivative [63]. The Caputo–Fabrizio (CF) derivative, introduced in 2015, replaces the singular power-law kernel with an exponential kernel:

$$
{}^{\mathrm{CF}}D_t^{\alpha} f(t) = \frac{M(\alpha)}{1-\alpha} \int_a^t f^{'}(\tau) \exp\left[-\frac{\alpha(t-\tau)}{1-\alpha} \right] d\tau \tag{5}
$$

where $M(\alpha)$ is a normalization function satisfying $M(0) = M(1) = 1$. The exponential kernel removes the singularity entirely, yielding a derivative that is more regular and better suited for problems where a "filtering" interpretation is appropriate. The CF derivative has been applied to the Allen–Cahn reaction-diffusion model, demonstrating its utility in capturing material heterogeneities and multi-scale structures. However, the CF derivative has drawn criticism: the associated integral operator is not a true fractional integral in the classical sense, and the derivative may be viewed more as a filter than as a genuine fractional operator [64]. The kernel's exponential decay implies that memory effects are of short-range character, limiting the operator's ability to capture long-term hereditary behavior typical of classical fractional derivatives.

Atangana–Baleanu derivative [65]. Responding to limitations of the CF formulation, Atangana and Baleanu [65] proposed a derivative employing the generalized Mittag-Leffler function as its kernel:

$$
{}^{\mathrm{AB}}D_{a+}^{\alpha} f(t) = \frac{B(\alpha)}{1-\alpha} \int_a^t f^{'}(\tau) E_{\alpha} \left[-\frac{\alpha(t-\tau)^{\alpha}}{1-\alpha} \right] d\tau, \tag{6}
$$

where $E_{\alpha}(z)$ denotes the one-parameter Mittag-Leffler function. The AB derivative combines two desirable properties: a non-singular kernel (eliminating the singularity at $t = \tau$) and a nonlocal kernel (preserving long-range memory through the Mittag-Leffler function's algebraic decay at infinity). This duality has made the AB derivative an attractive choice for diffusion and epidemic models that require strong memory effects without singular behavior [66]. Unlike the CF derivative, the AB operator admits a genuine fractional integral as its inverse, strengthening its mathematical standing as a fractional operator.

Conformable derivative [1]. The conformable fractional derivative, introduced by Khalil et al. [1], stands apart from all the operators discussed above by being fundamentally local in nature. Defined as:

$$T_\alpha f(x) = \lim_{\epsilon\to 0}\frac{f(x+\epsilon x^{1-\alpha})-f(x)}{\epsilon}, 0<\alpha<1. \quad (7)$$

The conformable derivative reduces to $T_\alpha f(x) = x^{1-\alpha}f'(x)$ for differentiable functions. This explicit relation to the classical derivative reveals the conformable derivative's essential character: it is not a genuinely nonlocal operator but rather a classical derivative weighted by a power of the independent variable. The conformable derivative inherits all classical properties, including the product rule, quotient rule, and chain rule—features absent from nonlocal fractional derivatives. This simplicity makes the conformable derivative computationally attractive and conceptually accessible [67,68]. However, precisely because it lacks the nonlocal memory effects that motivate fractional calculus, the conformable derivative does not capture the hereditary behavior that is the hallmark of fractional systems. Its application should be limited to problems where local behavior with a fractional-scaling factor is appropriate, rather than problems requiring genuine memory effects.

Two-scale fractal derivative [69]. For a function $u(t,x)$ defined on a fractal medium with fractal dimension α, the fractal derivative with respect to the fractal timescale t^α at the point (t_0,x) is defined as:

$$\frac{\partial u}{\partial t^\alpha}(t_0,x) = \Gamma(1+\alpha)\lim_{\substack{\Delta t\to\Delta t\\ \Delta t\neq 0}}\frac{u(t,x)-u(t_0,x)}{(t-t_0)^\alpha}, \quad (8)$$

where Δt is the smallest time scale characterizing the fractal structure, α is the fractal dimension ($0<\alpha<1$), and $\Gamma(1+\alpha)$ is the Gamma function serving as a normalization constant that ensures consistency with classical derivatives when $\alpha=1$.

This definition embodies several key insights: (i) Fractal scaling—the denominator $(t-t_0)^\alpha$ replaces the linear $(t-t_0)$ of classical derivatives; when $\alpha = 1$, the definition reduces to the standard first-order derivative (up to the normalization factor $\Gamma(2)=1$); for $0<\alpha<1$, the scaling is sublinear, reflecting the slower dynamics on fractal domains. (ii) Smallest time scale—the condition $\Delta t\to 0$ is understood in the sense of the smallest physically meaningful time scale, often the characteristic relaxation time of the fractal structure, preventing the limit from approaching the unphysical regime where continuum assumptions break down. (iii) Normalization constant—the inclusion of $\Gamma(1+\alpha)$ ensures that the fractal derivative of t^α equals 1, providing a natural analogy to the classical derivative of t.

The two-scale fractal derivative is a local, scale-transformation-based derivative used to model discontinuous, porous, unsmooth, or fractal-like media by separating behavior at a coarse scale and a fine scale . In contrast to the nonlocal Caputo-Fabrizio and Atangana-Baleanu operators discussed earlier, it does not rely on a memory kernel or convolution structure, and it has been widely applied to fractal vibration systems [70–74].

Among these definitions, the Caputo derivative remains the default choice for most physical and engineering applications due to its physically meaningful initial conditions and zero constant derivatives. The RL and GL derivatives are primarily of theoretical or numerical interest. The Riesz derivative is the standard choice for spatial fractional problems requiring symmetry. The CF and AB derivatives offer non-singular alternatives with short-range and long-range memory, respectively. The conformable derivative provides computational simplicity at the cost of genuine nonlocality, while the two-scale fractal derivative bridges fractional calculus with fractal geometry for problems on self-similar domains.

2.2. Anomalous diffusion and fractional-order control

The practical utility of fractional models is perhaps best illustrated by two paradigmatic application domains: anomalous diffusion and fractional-order control.

Many physical transport processes do not follow standard Fickian diffusion on molecular or micro/nano scales [75, 76], where the mean square displacement scales linearly with time as $\langle x^2(t)\rangle \sim t$. Instead, in porous media, amorphous semiconductors, polymeric structures, and biological tissues, the mean square displacement scales as $\langle x^2(t)\rangle \sim t^\alpha$ with $\alpha \neq 1$ [77]. When $\alpha < 1$, the process exhibits sub-diffusion, indicating slower spreading due to obstacles, traps, or binding effects; when $\alpha > 1$, super-diffusion occurs, reflecting faster spreading due to long-range jumps or Lévy flights. Fractional diffusion equations precisely capture this anomalous behavior, becoming standard tools in fluid mechanics, environmental science, and biophysics [78,79].

Fractional-order $PI^\lambda D^\mu$ controllers generalize classical proportional-integral-derivative controllers by allowing non-integer orders of integration (λ) and differentiation (μ). These fractional-order controllers demonstrate superior robustness, better anti-interference performance, and greater flexibility in shaping frequency responses compared to their integer-order counterparts, finding wide application in industrial process control, robotics, and power engineering [77].

2.3. Fractal-fractional models and their integration with PINNs

While the fractional derivatives surveyed above effectively capture memory effects and other nonlocal properties, they assume continuous and differentiable functions defined on standard, integer-dimensional domains. This assumption fails for systems evolving on fractal media—porous materials, rough surfaces, biological tissues, and geophysical formations—where the underlying domain itself possesses a non-integer fractal dimension. To address this limitation, the fractal-fractional derivative has been introduced, generalizing the Caputo, RL, CF, and AB definitions by replacing the ordinary derivative d/dt with the fractal derivative d/dt^β [80,81].

The resulting fractal-fractional formulations include:

Modified Caputo fractional derivative:

$$D_t^{\alpha,\beta} f(t) = \frac{1}{\Gamma(1-\alpha)} \int_0^t \frac{f^{(\beta)}(\tau)}{(t-\tau)^\alpha} d\tau. \tag{9}$$

Modified Riemann–Liouville derivative:

$$^{\mathrm{RL}}D_{a+}^{\alpha,\beta} f(t) = \frac{1}{\Gamma(n-\alpha)} \left(\frac{d}{dt^{\beta}} \right)^{n} \int_{a}^{t} \frac{f(\tau)}{(t-\tau)^{\alpha-n+1}} \, d\tau. \tag{10}$$

Modified Caputo–Fabrizio derivative:

$$^{\mathrm{CF}}D_{t}^{\alpha,\beta} f(t) = \frac{M(\alpha)}{1-\alpha} \int_{a}^{t} f^{(\beta)}(\tau) \exp\left[-\frac{\alpha(t-\tau)}{1-\alpha} \right] d\tau. \tag{11}$$

Modified Atangana–Baleanu derivative:

$$^{\mathrm{AB}}D_{a+}^{\alpha,\beta} f(t) = \frac{B(\alpha)}{1-\alpha} \int_{a}^{t} f^{(\beta)}(\tau) E_{\alpha} \left[-\frac{\alpha(t-\tau)^{\alpha}}{1-\alpha} \right] d\tau. \tag{12}$$

where β is the fractal dimension, and $f^{(\beta)}(\tau)$ denotes the fractal derivative with respect to t^{β}. These formulations bridge fractional calculus with fractal geometry, enabling more accurate modeling of systems with self-similar structures—such as porous catalysts, biological networks, and geophysical formations—where both the geometry of the domain and the dynamics of the process exhibit anomalous scaling [82].

Integrating fractal-fractional derivatives into PINNs poses two additional challenges beyond standard fPINNs: (i) the fractal derivative $f^{(\beta)}(\tau)$ itself requires numerical approximation at each historical grid point, compounding the $O(N^2)$ convolution cost; and (ii) the fractal dimension β enters as an additional unknown parameter, creating a twofold inverse problem that exacerbates ill-posedness and cross-coupling in the loss landscape. These challenges motivate the sensitivity-based identifiability diagnostics and staged training strategies discussed in Subsection 5.4 for multi-parameter fractional systems.

2.4. Implications for PINN-based modeling

The choice of fractional derivative definition directly impacts the fPINN architecture, loss function formulation, and training complexity. Classical definitions (RL, GL, Caputo) require $\mathcal{O}(N)$ operations per derivative evaluation due to the full history integral, leading to $\mathcal{O}(N^2)$ total complexity per training epoch. The CF and AB derivatives introduce exponential or Mittag-Leffler kernels that may be amenable to fast evaluation, though their integration with neural network training remains less explored. Riesz derivatives introduce spatial nonlocality, demanding dense collocation across the spatial domain with additional $\mathcal{O}(N^2)$ costs. Conformable derivatives impose no additional computational burden beyond classical derivatives, but their local nature makes them unsuitable for problems requiring genuine memory effects—the very motivation for fractional modeling. Fractal-fractional derivatives introduce the additional complexity of estimating the fractal dimension β alongside the fractional order α, representing a three-fold inverse problem when the solution $u(t)$ is also unknown.

For fPINN practitioners, the Caputo definition remains the most practical and widely validated choice for temporal fractional problems, while the Riesz derivative is recommended for spatial fractional diffusion. The AB and CF definitions represent emerging directions with theoretical appeal but less established integration with neural network frameworks. The fractal-fractional formulations are at the earliest stage of development and offer significant opportunities for future research at the intersection of fractional calculus, fractal geometry, and scientific machine learning.

3. Fundamentals of physics-informed neural networks

Physics-informed neural networks (PINNs) represent a paradigm shift in solving differential equations by embedding physical laws directly into the training process of neural networks [27]. This section introduces the architecture and training of standard PINNs, explains the role of automatic differentiation, and critically analyzes why the direct application of PINNs to fractional differential equations faces fundamental obstacles.

3.1. Architecture and training of standard PINNs

A standard PINN is built upon a fully connected feedforward neural network—typically a multilayer perceptron (MLP)—that approximates the solution to a given differential equation. Let the spatial and temporal variables be denoted as $\mathbf{x} \in \mathbb{R}^d$ and $t \in [0, T]$, with the unknown solution $u(\mathbf{x}, t)$ satisfying a governing differential equation $\mathcal{F}[u](\mathbf{x}, t) = 0$, subject to initial conditions $u(\mathbf{x}, 0) = u_0(\mathbf{x})$ and boundary conditions $\mathcal{B}[u](\mathbf{x}, t) = 0$ on $\partial\Omega \times [0, T]$. The neural network takes $(\mathbf{x}, t)$ as input and outputs the approximation $\widehat{u}(\mathbf{x}, t; \theta)$, where θ denotes the set of trainable parameters—weights and biases—to be optimized.

The network architecture typically consists of an input layer, multiple hidden layers with nonlinear activation functions (e.g., tanh, ReLU, or swish), and an output layer. The universal approximation theorem guarantees that sufficiently wide neural networks can approximate any continuous function to arbitrary accuracy, providing the theoretical foundation for PINN-based solution approximations. In practice, the depth and width of the network are chosen based on problem complexity, with deeper networks generally capable of representing more complex solution landscapes at the cost of increased training difficulty.

3.2. Automatic differentiation and its role in PINNs

The key enabling technology for PINNs is automatic differentiation (AD). Unlike numerical differentiation (which suffers from truncation and round-off errors) or symbolic differentiation (which is infeasible for complex composite functions), AD computes derivatives of the network output with respect to its inputs exactly up to machine precision. Through the systematic application of the chain rule—implemented via computational graphs in forward or reverse mode—AD yields exact integer-order derivatives such as $\partial\widehat{u}/\partial t$, $\partial^2\widehat{u}/\partial x^2$, and higher-order combinations.

For a standard (integer-order) differential equation $\mathcal{F}[u] = 0$, the physics residual is constructed by substituting the network approximation $\widehat{u}$ into the governing equation:

$$\mathcal{R}(\mathbf{x}, t; \theta) = \mathcal{F}[\widehat{u}](\mathbf{x}, t; \theta). \tag{13}$$

All derivatives appearing in $\mathcal{F}$—including first-order time derivatives, second-order spatial derivatives, and mixed partial derivatives—are computed via AD. This capability allows PINNs to enforce the governing equation at a set of collocation points without requiring mesh generation or explicit discretization schemes. The mesh-free nature of PINNs, combined with the accuracy of AD, constitutes their principal advantages over traditional numerical solvers.

3.3. Loss function formulation and optimization

The total loss function in a standard PINN combines three components: the physics loss (enforcing the governing equation), the data loss (matching observations), and the initial/boundary loss (satisfying constraints). Mathematically:

$$\mathcal{L}_{\text{total}}(\theta) = \lambda_p \mathcal{L}_{\text{physics}}(\theta) + \lambda_d \mathcal{L}_{\text{data}}(\theta) + \lambda_{ic} \mathcal{L}_{\text{IC}}(\theta) + \lambda_{bc} \mathcal{L}_{\text{BC}}(\theta), \tag{14}$$

where:

$$\mathcal{L}_{\text{physics}}(\theta) = \frac{1}{N_c} \sum_{j=1}^{N_c} |\ \mathcal{F}[\widehat{u}](\mathbf{x}_j, t_j; \theta)\ |^2, \tag{15}$$

$$\mathcal{L}_{\text{data}}(\theta) = \frac{1}{N_d} \sum_{i=1}^{N_d} |\ \widehat{u}(\mathbf{x}_i, t_i; \theta) - u_i\ |^2, \tag{16}$$

$$\mathcal{L}_{\text{IC}}(\theta) = \frac{1}{N_{ic}} \sum_{k=1}^{N_{ic}} |\ \widehat{u}(\mathbf{x}_k, 0; \theta) - u_0(\mathbf{x}_k)\ |^2, \tag{17}$$

$$\mathcal{L}_{\text{BC}}(\theta) = \frac{1}{N_{bc}} \sum_{l=1}^{N_{bc}} |\ \mathcal{B}[\widehat{u}](\mathbf{x}_l, t_l; \theta)\ |^2. \tag{18}$$

Here, λ_p, λ_d, λ_{ic}, and λ_{bc} are weighting coefficients that balance the contributions of each loss component; N_c, N_d, N_{ic}, and N_{bc} denote the number of collocation points, data points, initial condition points, and boundary condition points, respectively. The optimization problem seeks the parameter set θ^* that minimizes the total loss:

$$\theta^* = \arg\min_\theta \mathcal{L}_{\text{total}}(\theta). \tag{19}$$

The minimization is typically performed using gradient-based optimizers—most commonly Adam for initial exploration, followed by L-BFGS for fine-tuning. The gradients $\nabla_\theta \mathcal{L}_{\text{total}}$ are computed via backpropagation, leveraging the same AD framework used for derivative computation.

3.4. Why standard PINNs fail for fractional differential equations

Standard PINNs, despite their remarkable success for integer-order differential equations, cannot be directly applied to fractional differential equations. The fundamental reason lies in the core assumption of PINN training: all derivatives are computed via automatic differentiation using the classical chain rule. Fractional derivatives, however, are nonlocal operators that do not obey the classical chain rule. As Wang et al. [54] and Srati et al. [83] emphasized, standard PINNs must be modified with numerical discretization, transform methods, or specialized architectures to handle fractional operators correctly.

Three fundamental obstacles arise:

Obstacle 1: Non-locality incompatibility [84]: AD computes local derivatives $\partial^n u/\partial x^n$ that depend only on the function's behavior in an infinitesimal neighborhood of the evaluation point. In contrast, the Caputo fractional derivative in Equation (1) involves a convolution integral over the entire history from 0 to t. That is the value of $D_t^\alpha f(t)$ depends on $f(\tau)$ for all $\tau \in [0, t]$—not just the local behavior at t. This nonlocal dependence cannot be captured by AD, which only tracks local derivatives. Attempting to apply AD directly to a fractional derivative would require the network to "remember" its own outputs at all previous time points, a capability that standard feedforward architectures do not possess.

Obstacle 2: Approximation necessity: Since many standard fractional derivatives exhibit nonlocal behavior and often lack universally applicable chain-rule-compatible formulations, they are frequently approximated numerically using discretization frameworks including Grünwald–Letnikov series, the L1 scheme, convolution quadrature, and spectral methods to resolve their underlying convolution integrals—though it is critical to note numerical approximation is not the sole viable approach, as specialized fractional operators admit rigorous analytical treatments and dedicated chain-rule identities. Every such numerical discretization generates inherent truncation and discretization errors, which interact in complex, non-trivial manners with the approximation error originating from neural network fitting [85, 86]. Beyond this, the selection of a fractional derivative discretization scheme and its associated hyperparameters (such as the temporal step size Δt) exert a substantial influence on the overall accuracy and numerical stability of fractional physics-informed neural networks (fPINNs), introducing an extra set of tunable hyperparameters absent from conventional integer-order PINNs.

Obstacle 3: Computational overhead: Each evaluation of a fractional derivative at a single time point requires $\mathcal{O}(N)$ operations, where N is the number of historical discretization points. For N_c collocation points distributed over the temporal domain, the total cost scales as $\mathcal{O}(N_c \cdot N)$, or $\mathcal{O}(N^2)$ when collocation points coincide with the temporal grid [87]. This quadratic complexity stands in stark contrast to standard PINNs, where derivative evaluation via AD is $\mathcal{O}(1)$ per point. For problems with fine temporal resolution (e.g., $N > 10^4$), the computational cost becomes prohibitive for standard GPU hardware.

To quantitatively guide readers in selecting appropriate fractional discretization schemes when building fPINN models, we summarize the implementation pipeline, training differences, and practical application status of three mainstream approximation methods in **Table 1**, focusing on their coupling logic with the PINN automatic differentiation framework rather than only introducing pure numerical algorithms.

As summarized in **Table 1**, GL and L1 schemes dominate existing fractional PINN implementations due to their natural compatibility with mesh-free collocation and automatic differentiation. In contrast, spectral approximation methods face inherent incompatibility with the core architecture of standard PINNs, leading to extremely limited practical applications in general fractional forward and inverse problems. Subsequent subsections will further detail the specific loss construction of GL and L1-based fPINNs.

Table 1. Comparison of mainstream fractional derivative approximation schemes for fractional PINNs.

Comparison Dimension	Grünwald–Letnikov (GL)	L1 scheme (Caputo-adapted)	Spectral method
Embedding Pipeline in fPINNs	1. Discretize the time domain into historical grid points at each collocation node; 2. Calculate fractional difference weights via binomial coefficients; 3. Substitute weighted historical network outputs into the fractional derivative term of the physics residual; 4. Combine with AD to compute loss gradients for network optimization.	1. Precompute fixed integral weights for the piecewise power-law kernel of Caputo derivative; 2. Load the pre-stored weight matrix to calculate fractional derivative values at collocation points; 3. Integrate fractional residual into total loss together with IC/BC/data loss; 4. Backpropagate loss gradients via AD jointly with fractional discretization weights.	1. Approximate solution via global orthogonal polynomial basis expansion over the entire domain; 2. Compute fractional operators using spectral differentiation matrix; 3. Conflicts with the mesh-free collocation paradigm of standard PINNs, requiring full global grid discretization; 4. Cannot natively couple with local AD of feedforward networks.
Training & Gradient Characteristics	Differentiable weights support end-to-end backpropagation; training is sensitive to the time step Δt and the historical truncation window; long memory windows induce quadratic computational cost.	Higher accuracy for smooth time series than GL; stable gradient landscape under noisy sparse data; offline weight calculation reduces online training overhead.	Global full-domain coupling causes massive memory consumption; gradient computation involves all domain points simultaneously, which can easily trigger training instability.
Practical Adoption in fPINN Literature	Dominant mainstream scheme; most open-source fPINN baselines and engineering implementations adopt GL + AD coupling.	Widely preferred for time-fractional initial value problems; standard choice for Caputo-type fractional systems in fPINN inverse identification.	Extremely rare in general fPINN implementations; almost no open-source fPINN frameworks support spectral embedding; only limited to theoretical numerical analysis.
Core Limitations & Applicable Scenarios	Limitation: Truncation error exists for long time intervals; Scenario: General time-fractional PDEs, low-to-medium time-resolution simulation.	Limitation: Only suitable for Caputo fractional derivatives; Scenario: Viscoelastic modeling, anomalous subdiffusion, fractional order identification tasks.	Limitation: Incompatible with mesh-free PINN core features; poor adaptability to irregular domains and complex boundary conditions; Scenario: Ideal periodic regular domains for pure theoretical comparison only; not recommended for engineering fPINN modeling.

The foundational work of fPINNs was established by Pang et al. in 2019 [88]. Since then, the technical evolution of fractional PINNs has witnessed a clear paradigm shift: the initial hybrid residual framework has gradually evolved into a diverse suite of specialized variants that deliver improved numerical accuracy, better computational scalability, and more stable training dynamics, while extending the applicability of fPINNs to complex, intractable fractional differential problems.

Building upon the core fPINN architecture, Pang et al. further generalized the nonlocal neural solver framework to propose nPINNs for parametrized nonlocal Laplacian-type operators, alongside other fractional neural approximations such as FCDNN [89]. This line of research reveals a rapidly expanding research interest in neural network solvers for general nonlocal and fractional PDEs, moving beyond the original time-fractional fPINN prototype.

1. **Scalability and high-dimensional extension [90,91]**

   Two mainstream technical routes have been proposed to improve the scalability of fractional PINNs (fPINNs) for large-scale fractional partial differential equation problems. The first is Monte Carlo fPINNs, which substantially reduce computational costs by approximating fractional integrals through stochastic sampling [90]. The second category covers spectral-collocation-enhanced fPINNs, which significantly elevate training efficiency and pointwise prediction accuracy for time-fractional phase-field systems [84, 91]. As elaborated in the 2022 study [84], this paradigm integrates spectral collocation techniques into the standard fPINN framework: it cuts the quantity of auxiliary approximation points required for fractional operator discretization, lowers the computational cost of matrix operations, and attains pointwise numerical errors ranging from 10^{-5} to 10^{-7} in numerical tests.

2. **Accuracy-oriented improved architectures [92–94]**
   A series of refined fPINN variants are proposed to elevate fitting precision by introducing tailored network structures or extra physical constraints. This category includes basic improved fractional PINNs, physical-model-driven neural networks (PMNNs), and bi-orthogonal fPINNs designed for stochastic fractional forward and inverse problems [92–94].
3. **Efficiency and adaptive optimization innovations [95–97]**
   Recent methodological innovations focus on alleviating the inherent quadratic complexity of fractional operators and enhancing training robustness. Representative strategies include Monte Carlo/quasi-Monte Carlo integral approximation, Laplace transform domain decoupling, gradient-enhanced residual loss, adaptive loss weighting schemes, adjustable learning rates, and adaptive activation functions, as well as Hermite/spectral basis decomposition. Collectively, these techniques strengthen model convergence and dimensional adaptability for challenging fractional inverse tasks [52, 95–97].

3.5. Strategies for overcoming the obstacles

The obstacles identified above have motivated three mainstream adaptation strategies.

3.5.1. Simultaneous identification for single fractional orders

This strategy keeps the standard PINN architecture but replaces integer-order treatment of fractional operators with numerical fractional approximations; in the foundational fPINN formulation, Pang et al. used automatic differentiation for integer-order terms and numerical discretization for fractional terms [89].

3.5.2. Network-structure-enhanced fractional PINNs

Different from fractional-order modified PINNs that only update the derivative calculation rules, network-structure-enhanced fractional PINNs optimize the internal feature extraction capability of neural networks. It is necessary to clarify that the core architectural improvements adopted in this branch, including residual connections, multi-scale branches, and attention mechanisms, are universal network optimization techniques borrowed from general deep learning fields, rather than original structures specially designed for fractional differential systems. Nevertheless, these generic architectures have unique adaptive values for solving fractional differential equations (FDEs) and can effectively address the core challenges of fractional PINNs. Specifically, the non-local and memory-dependent properties of fractional derivatives require networks to perceive long-range temporal and spatial correlation information, which cannot be well captured by standard multilayer perceptrons. Residual connections solve the gradient degradation problem in deep network training for long-time fractional dynamic fitting; multi-scale branching structures adapt to the multi-scale anomalous diffusion and fractal scaling characteristics of fractional systems; spatial-temporal attention mechanisms dynamically assign reasonable weights to historical memory terms of fractional operators, filtering invalid redundant information and significantly improving the stability and fitting accuracy of non-local fractional residual calculation [90].

3.5.3. Hybrid methods

It combines traditional numerical solvers (e.g., finite difference for spatial fractional terms) with PINNs for temporal evolution, balancing efficiency and stability [98,99].

The remainder of this review focuses on these strategies, their mathematical formulations, and their applications to fractional differential equations.

3.5.4. Regularization for Coupled Multi-Order Systems

It keeps the standard PINN architecture but replaces the integer-order treatment of fractional operators with numerical fractional approximations. A particularly effective enhancement is the homotopy-based loss formulation, which interpolates between the integer-order and fractional-order governing equations:

$$\mathcal{L}_{physics}^{hom}(p,\theta) = \frac{1}{N_c}\sum\nolimits_{j=1}^{N_c} \mid (1-p)[\dot{u}_\theta(t_j) + \mathcal{N}[u_\theta](t_j)] + p[D_t^\alpha u_\theta(t_j) + \mathcal{N}[u_\theta](t_j)] \mid^2. \tag{20}$$

By scheduling p from 0 to 1 during training, the network first learns the easier integer-order dynamics via stable automatic differentiation, then gradually accommodates the non-local fractional effects as p increases. This curriculum-learning strategy substantially improves training stability and convergence for ill-conditioned fractional inverse problems.

4. Three mainstream branches of fractional PINNs

Standard PINNs fail for fractional differential equations fundamentally due to the non-local convolution integral of fractional derivatives, which cannot be directly computed via chain-rule-based automatic differentiation. Based on distinct strategies to resolve this core conflict, existing fractional PINN (fPINN) methods are categorized into three mutually exclusive branches: (i) fractional-order modified PINNs, which retain standard MLP backbones and embed discrete fractional approximation formulas to reconstruct physical residuals; (ii) network-structure-enhanced fractional PINNs, which upgrade network architectures with residual connections, multi-scale branches, and attention mechanisms to capture long-range memory correlations; and (iii) hybrid numerical-PINN methods, which couple traditional grid-based fractional solvers with neural networks to separate spatial and temporal fractional computation tasks. This taxonomy is not a simple catalog of literature but an analytical framework built around the theoretical and computational bottlenecks of fractional operators.

The first branch—fractional-order modified PINNs—preserves the fully connected network backbone of original PINNs, concentrating all improvements on rewriting fractional derivative terms in the loss function. Three dominant discretization schemes—Grünwald–Letnikov (GL), L1, and spectral methods—are widely used to approximate Caputo, Riemann–Liouville, and Riesz fractional derivatives. The GL scheme adopts binomial difference weights and supports end-to-end backpropagation, making it the dominant mainstream choice in most open-source fPINN baselines, though it suffers from first-order convergence and strict CFL-like time-step restrictions. The L1 scheme, which precomputes fixed integral weights for the Caputo kernel, delivers second-order accuracy and stable gradient landscapes, making it the standard choice for time-fractional initial value problems and fractional-order identification tasks. Spectral methods achieve exponential convergence for smooth periodic functions on regular domains but conflict with the mesh-free collocation paradigm of standard PINNs, requiring full global grid discretization that is incompatible with local automatic differentiation; they are therefore rarely adopted in general fPINN implementations. This branch requires minimal modification to standard PINN codes and performs stably for low-dimensional, smooth time-fractional forward problems with abundant clean collocation data, but all discretization strategies introduce truncation error coupled with neural network approximation error, and the full-history summation of fractional operators brings unavoidable quadratic complexity that drastically increases training overhead for long-time dynamic modeling.

The second branch—network-structure-enhanced fractional PINNs—addresses the same non-local obstacle from a different perspective. Rather than merely revising residual calculations, this branch fixes mature fractional discretization modules and improves the network's capacity to extract long-range temporal and spatial correlations. It is important to clarify that residual connections, multi-scale branches, and spatial-temporal attention modules adopted here are generic deep learning techniques originating from computer vision and sequence modeling, rather than dedicated innovations for fractional differential equations. These structures are adaptively introduced to compensate for standard MLPs' weakness in capturing non-local memory effects induced by fractional operators. Residual layers alleviate gradient vanishing when fitting long-time fractional dynamics; multi-scale branches match the multi-scale anomalous scaling characteristics of fractal-fractional systems; attention layers dynamically assign weights to historical memory terms and filter redundant temporal information, demonstrating unique advantages in fractional-order identification and variable-order inverse problems. However, these extra layers greatly expand network parameter volume and training memory occupation, and no quantitative theoretical relation exists between attention weight distribution and fractional kernel decay laws. The optimization effect heavily relies on manual tuning of network depth, branch number, and attention head count, and generalization ability degrades sharply under extremely sparse and noisy measurement data.

The third branch—hybrid numerical-PINN methods—takes a fundamentally different systems-level approach. These frameworks combine traditional grid-based fractional numerical solvers (finite difference, finite element) with neural networks, separating spatial fractional diffusion calculation and temporal nonlinear evolution approximation to balance computational efficiency and fitting accuracy. Classical discrete grids handle spatial non-local fractional terms, while mesh-free PINNs approximate time-dependent dynamic responses on complex unstructured domains. Offline precomputed grid weight matrices reduce online training computation, and classical numerical discretization provides extra physical constraints to suppress overfitting under limited observation data. This branch shows outstanding adaptability to high-dimensional spatial fractional PDEs and irregular geometric domains. Nevertheless, grid discretization reintroduces the mesh generation burden that vanilla mesh-free PINNs aim to eliminate, and complex data interaction between numerical solvers and neural networks breaks the automatic differentiation flow, requiring customized reconstruction of coupling logic for different boundary condition forms.

Despite their distinct technical pathways, all three branches share two universal unsolved core bottlenecks: the quadratic computational overhead originating from full-history fractional convolution integrals, and incomplete rigorous theoretical guarantees for joint convergence and parameter identifiability of fPINN systems. The practical choice among them should be guided by problem characteristics rather than a universal preference. For lightweight low-dimensional forward modeling with adequate high-quality data, fractional-order modified PINNs deliver the best balance of simplicity and computational cost. For memory-sensitive inverse problems such as fractional parameter estimation, network-structure-enhanced architectures present superior fitting capacity. For high-dimensional spatial fractional models with limited sampling points on irregular domains, hybrid numerical-PINN methods obtain the most stable prediction results. **Table 1** provides a comprehensive comparison of the three mainstream fractional discretization schemes (GL, L1, spectral) in terms of embedding pipeline, training characteristics, practical adoption status, and core limitations. Collectively, these three branches address the non-local obstacle from completely independent technical perspectives, and no single framework achieves optimal performance across all fractional differential tasks, leaving the two critical limitations—computational complexity and theoretical guarantees—as the most valuable research breakthrough directions for subsequent work.

5. Identification of fractional orders

Fractional orders are core characteristic parameters of fractional differential equations, governing the intensity of memory effects and the extent of non-local correlations in dynamical systems. In many practical applications—such as viscoelastic material characterization, epidemic modeling, and fractional-order controller design—the fractional order α cannot be measured directly and must be inferred inversely from limited observational data. This inverse problem, known as fractional-order identification, poses inherent numerical difficulties: fractional orders enter the derivative operators nonlinearly and are tightly coupled with the unknown solution field, resulting in a strongly ill-posed joint estimation task. Before presenting the complete mathematical framework for fractional-order identification based on fractional discretization schemes, we first systematically introduce three mainstream auxiliary strategies for constructing fractional residuals within fractional physics-informed neural networks (fPINNs). These include fractional discretization, Laplace transform operators, and Monte Carlo stochastic sampling approximations, each tailored to different fractional modeling requirements. Since Section 4 has already provided a comprehensive discussion of the Grünwald–Letnikov (GL) and L1 fractional discretization methods, this section focuses on the remaining two alternative schemes, detailing their core mathematical principles, specific embedding workflows in PINNs, inherent strengths and limitations, as well as representative landmark studies.

The Laplace transform fundamentally addresses the $O(N^2)$ computational bottleneck of GL and L1 discretization, which arises from full-history convolution summation. By converting non-local fractional integral operators defined in the time domain into simple algebraic expressions in the complex frequency domain, this approach completely eliminates cumulative memory summation terms. For a time-fractional Caputo-type governing equation $D_t^\alpha u = N(u)$, the Laplace transform of the fractional derivative satisfies $\mathcal{L}\{D_t^\alpha u\} = s^\alpha U(s) - \sum_{k=0}^{n-1} s^{\alpha-k-1} u^{(k)}(0)$, which transforms the fractional differential operator into a power function of the complex frequency s and entirely removes the integral memory kernel. When embedding this mechanism into PINNs, researchers first construct a time-domain surrogate neural network $u_\theta(t)$ to approximate the unknown solution, then map the network outputs into the Laplace domain via numerical Laplace transformation. The fractional derivative term in the physics residual is subsequently replaced by its algebraic frequency-domain counterpart, bypassing the need for discretized historical time grids. Finally, an inverse Laplace transform loss term is appended to enforce consistency between frequency-domain predictions and time-domain physical constraints, and all network weights are jointly optimized via gradient descent. This scheme offers two prominent advantages: it avoids temporal discretization truncation errors and maintains linear computational complexity, demonstrating outstanding performance for linear time-fractional partial differential equations with smooth initial conditions. Nevertheless, it has obvious limitations: numerical inverse Laplace transformation introduces additional approximation errors, and the framework is difficult to generalize to coupled space–time fractional systems or variable-order fractional models. A representative example of this technical route is the Laplace-PINN proposed by Wang et al. [54] for transient fractional diffusion simulation, whose numerical experiments validated superior computational efficiency over the L1 discretization scheme in long-time dynamic evolution cases.

In contrast to the frequency-domain transformation underlying Laplace-based PINNs, Monte Carlo (MC) approximation reconstructs fractional convolution integrals through stochastic expectation estimation. Rather than performing deterministic summation over all historical grid points, this method employs randomly sampled subsets to approximate integral terms, significantly reducing computational costs for high-dimensional systems defined on irregular domains, with GMC-PINN serving as a classic implementation. Mathematically, the fractional integral $\int_0^t (t-\tau)^{\alpha-1} u(\tau) d\tau$ can be reformulated as the mathematical expectation of random variables following a power-law distribution; only a small batch of stochastic samples is needed to approximate the integral value, eliminating the need for full traversal of historical collocation nodes. In the specific PINN embedding process, practitioners first randomly select a sparse subset of historical time nodes instead of using uniform full temporal grids, then compute fractional derivative values via weighted Monte Carlo expectation estimates, and finally incorporate a sampling variance regularization term into the total loss function to suppress stochastic fluctuations during backpropagation training. In terms of practical performance, Monte Carlo sampling PINNs drastically reduce computational costs for high-dimensional fractional models on irregular domains and exhibit good compatibility with spatial Riesz fractional operators. However, the random sampling mechanism inevitably induces persistent oscillation in loss values, requiring multiple repeated sampling rounds to stabilize training, and the method yields unsatisfactory prediction accuracy for short-time, low-resolution simulations. Wang and Karniadakis [52] developed the GMC-PINN framework following this rationale, achieving better accuracy and efficiency than standard GL-fPINN in solving multi-dimensional anomalous fractional diffusion models on complex computational domains.

Combining the GL and L1 fractional discretization schemes elaborated in Section 4, we further distill targeted selection criteria for the three types of operator-aided PINN strategies, thereby offering actionable guidance for fPINN practitioners. Fractional discretization methods, represented by GL and L1 schemes, are most suitable for low-dimensional constant-order time-fractional systems and have become the most widely adopted technical framework for fractional-order identification research. Laplace transform-based PINNs are the preferred option for linear long-time-fractional PDEs without complex spatial boundaries or variable-order characteristics. Monte Carlo sampling PINNs stand out as the optimal choice for high-dimensional spatial fractional equations and fractional models defined on irregular geometric domains. With these three core residual construction strategies fully clarified, we proceed to establish a complete mathematical formulation for fractional-order identification within the fPINN framework. The subsequent discussion provides an in-depth analysis of simultaneous joint optimization and two-stage sequential identification strategies, and further extends the investigation to multi-constant-order and space–time variable-order fractional dynamical systems.

5.1. Mathematical formulation of the order identification problem

Consider a fractional differential equation of the general form:

$$D_t^{\alpha} u(t) = \mathcal{N}[u](t), t \in [0, T], \tag{21}$$

with initial condition $u(0) = u_0$, where $\mathcal{N}$ is a (possibly nonlinear) differential operator and $0 < \alpha < 1$ is the unknown fractional order to be identified. Given discrete noisy observations $\left\{(\widehat{t}_i, \widehat{u}_i)\right\}_{i=1}^{N_d}$ of the system state, the task is to simultaneously estimate both the solution $u(t)$ and the fractional order α. This is a coupled inverse problem: the solution depends on the order, and the order must be inferred from the solution's behavior.

A fundamental insight, rooted in the variational iteration method (VIM), provides a unified perspective on the structure of fractional operators and their identification. The

VIM framework characterizes the fractional derivative operator through a variational formulation that naturally separates the fractional integral kernel from the differentiable part of the function. Specifically, the fractional derivative can be expressed in terms of an integral operator that encapsulates the nonlocal memory structure:

The variational iteration method provides an effective approach to characterize and formulate the definitions of various fractional derivatives [100]. Consider a linear differential equationL

$$u^{(n)} = f(t). \tag{22}$$

Its variational iteration solution is :

$$u(t) = u_0(t) + (-1)^n \int_{t_0}^{t} \frac{1}{(n-1)!}(s-t)^{n-1}\left[u_0^{(n)}(s) - f(s)\right] \mathrm{d}s. \tag{23}$$

So we can introduce an operator I defined as:

$$I^n f = \int_{t_0}^{t} \frac{1}{(n-1)!}(s-t)^{n-1}\left[u_0^{(n)}(s) - f(s)\right] \mathrm{d}s = \frac{1}{\Gamma(n)} \int_{t_0}^{t} (s-t)^{n-1}\left[f_0(s) - f(s)\right] \mathrm{d}s. \tag{24}$$

Where $f_0(t) = u_0^{(n)}(t)$. So we have [82]:

$$D_t^{\alpha} f = D_t^{\alpha} \frac{d^n}{dt^n}(I^n f) = \frac{d^n}{dt^n}(I^{n-\alpha} f) = \frac{1}{\Gamma(n-\alpha)} \frac{d^n}{dt^n} \int_{t_0}^{t} (s-t)^{n-\alpha-1}\left[f_0(s) - f(s)\right] ds, \tag{25}$$

where I^n denotes the n-fold integral operator, and $f_0(s)$ represents an initial guess or reference trajectory. This formulation reveals that the fractional order α appears explicitly in two places: (i) in the Gamma function $\Gamma(n-\alpha)$ as a normalization factor, and (ii) in the exponent of the kernel $(s-t)^{n-\alpha-1}$[101–103]. The VIM representation thus provides a principled basis for constructing loss functions that isolate the order dependence, facilitating gradient-based optimization for α identification—a perspective that has informed several PINN-based inverse frameworks [104–107].

In the PINN framework, we construct a neural network $u_\theta(t)$ with parameters θ (weights and biases). The fractional order α is treated as an additional trainable parameter, allowing the network to adapt both the solution and the order simultaneously to match the observations.

5.2. Simultaneous identification strategy

A fundamental difficulty in simultaneous fractional-order identification is the coupling between the unknown order α and the network parameters θ: inaccurate α early in training generates erroneous fractional residuals, which in turn corrupt the solution approximation $u_\theta(t)$, leading to oscillatory convergence or premature trapping in local minima. To mitigate this issue, we introduce a homotopy-based loss formulation that interpolates between the integer-order and fractional-order governing equations during training.

Consider a general fractional differential equation:

$$D_t^{\alpha} u(t) + \mathcal{N}[u](t) = 0, t \in [0, T], 0 < \alpha < 1. \tag{26}$$

Instead of directly minimizing the fractional residual $\| \ D_t^{\alpha} u_\theta + \mathcal{N}[u_\theta] \ \|^2$ throughout training, we define a homotopy residual parameterized by a continuation parameter $p \in [0, 1]$:

$$\mathcal{R}_{hom}(p, \theta, \alpha; t) = (1 - p) \left[\dot{u}_\theta(t) + \mathcal{N}[u_\theta](t)\right] + p \left[D_t^{\alpha} u_\theta(t) + \mathcal{N}[u_\theta](t)\right],$$

where $\dot{u}_\theta$ denotes the first-order time derivative computed via automatic differentiation. The homotopy physics loss is then:

$$\mathcal{L}_{physics}^{hom}(p, \theta, \alpha) = \frac{1}{N_c} \sum_{j=1}^{N_c} \mid \mathcal{R}_{hom}(p, \theta, \alpha; t_j) \mid^2, \tag{27}$$

and the total loss becomes:

$$\mathcal{L}_{total}^{hom}(p, \theta, \alpha) = \mathcal{L}_{data}(\theta) + \lambda_{hom} \mathcal{L}_{physics}^{hom}(p, \theta, \alpha) + \mathcal{L}_{IC}(\theta). \tag{28}$$

The continuation parameter p is scheduled from 0 to 1 over the course of training. When $p = 0$, the physics loss enforces the integer-order equation $\dot{u}_\theta + \mathcal{N}[u_\theta] = 0$, for which derivatives are computed exactly and stably via automatic differentiation. As p increases, the fractional contribution is gradually introduced, allowing the network to first capture the coarse-grained integer-order dynamics before accommodating the non-local memory effects. At $p = 1$, the homotopy loss recovers the original fractional residual exactly.

This homotopy strategy delivers three concrete benefits for fractional-order identification:

- **(i) Stabilized early-stage training.** In the initial phase (p near 0), the loss landscape is dominated by integer-order residuals that are smooth and convex locally, enabling stable gradient descent and rapid reduction of the data misfit. The fractional order α has minimal influence at this stage, preventing premature commitment to a poor estimate.
- **(ii) Graduated $\boldsymbol{\alpha}$ sensitivity**. As p increases, the sensitivity $\partial \mathcal{L}_{total} / \partial \alpha$ grows progressively from zero to its full magnitude. This avoids the abrupt onset of large, noisy gradients with respect to α that typically destabilizes simultaneous identification in standard fPINNs.
- **(iii) Improved multi-order identifiability.** For systems with multiple fractional orders $\alpha_1, \alpha_2, \ldots, \alpha_m$, the homotopy path can be designed to activate different orders at different rates, effectively decoupling their individual contributions to the loss. For instance, one may set $p_k(t)$ for order α_k with staggered schedules, allowing the network to identify dominant orders before finer ones.

The scheduling of p can follow several strategies. The simplest is linear scheduling:

$$p_k = \min\left(1, \frac{k}{K_{trans}}\right), k = 1, 2, \ldots, K_{total}, \tag{29}$$

where K_{trans} is the transition step at which p reaches 1. Alternative strategies include exponential scheduling $p_k = 1 - \exp(-\gamma k / K_{trans})$ for faster early progression, and adaptive scheduling that monitors the reduction of $\mathcal{L}_{data}$ and increments p only when the data loss has sufficiently decreased, ensuring that the integer-order dynamics are adequately learned before fractional effects are introduced.

For the two-stage sequential identification strategy (Subsection 5.3), homotopy can be applied in Stage 2 when optimizing α with the frozen network $u_{\theta^*}(t)$:

$$\alpha^* = \arg\min_{\alpha\in(0,1)} \frac{1}{N_c}\sum_{j=1}^{N_c} \mid (1-p)[\dot{u}_{\theta^*}(t_j)+\mathcal{N}[u_{\theta^*}](t_j)] + p[D_t^\alpha u_{\theta^*}(t_j)+\mathcal{N}[u_{\theta^*}](t_j)] \mid^2. \tag{30}$$

In this decoupled setting, homotopy serves as a regularization path that gradually introduces fractional complexity, making the reduced one-dimensional optimization over α more robust to local minima.

A practical guideline is that for ill-conditioned inverse problems with noise level $\sigma_{data} > 0.05$ or for multi-order systems with closely spaced orders ($\mid \alpha_i - \alpha_j \mid < 0.15$), homotopy-enhanced simultaneous identification often outperforms both standard simultaneous training and even the two-stage sequential approach in terms of final estimation accuracy, at the cost of approximately 20–30% additional computational overhead due to the multi-stage training schedule.

In practice, the simultaneous identification loss can be further enhanced by a homotopy continuation strategy. Replacing the standard physics loss with its homotopy counterpart yields:

$$\begin{gathered}\mathcal{L}_{total}^{hom}(p,\theta,\alpha) = \mathcal{L}_{data}(\theta) + \\ \lambda_{hom}\tfrac{1}{N_c}\textstyle\sum_{j=1}^{N_c} \mid (1-p)[\dot{u}_\theta(t_j)+\mathcal{N}[u_\theta](t_j)] + p[D_t^\alpha u_\theta(t_j)+\mathcal{N}[u_\theta](t_j)] \mid^2 \\ +\mathcal{L}_{IC}(\theta),\end{gathered} \tag{31}$$

with p scheduled from 0 to 1 during training. This formulation stabilizes early-stage optimization by initially emphasizing integer-order dynamics before gradually introducing fractional nonlocality.

The most direct approach to order identification treats α as a global trainable parameter alongside the network weights. The composite loss function becomes:

$$\mathcal{L}(\theta,\alpha) = \mathcal{L}_{\text{data}}(\theta) + \mathcal{L}_{\text{physics}}(\theta,\alpha) + \mathcal{L}_{\text{IC}}(\theta), \tag{32}$$

where:

$$\mathcal{L}_{\text{data}}(\theta) = \frac{1}{N_d}\sum_{i=1}^{N_d} \mid u_\theta(\widehat{t_i}) - \widehat{u}_i \mid^2, \tag{33}$$

$$\mathcal{L}_{\text{physics}}(\theta,\alpha) = \frac{1}{N_c}\sum_{j=1}^{N_c} \mid D_t^\alpha u_\theta(t_j) - \mathcal{N}[u_\theta](t_j) \mid^2, \tag{34}$$

$$\mathcal{L}_{\text{IC}}(\theta) = \mid u_\theta(0) - u_0 \mid^2. \tag{35}$$

The fractional derivative term $D_t^\alpha u_\theta(t_j)$ is computed using one of the approximation schemes from Section 4 (e.g., Grünwald–Letnikov, L_1 scheme, or spectral method), where the order α explicitly appears in the coefficients—for instance, $\binom{\alpha}{k}$ in the GL scheme, or $\Gamma(2-\alpha)$ and the weights $b_j = (j+1)^{1-\alpha} - j^{1-\alpha}$ in the L_1 scheme.

The total loss $\mathcal{L}(\theta,\alpha)$ is minimized via gradient descent, updating both network weights and α synchronously:

$$\theta \leftarrow \theta - \eta_\theta \nabla_\theta \mathcal{L}(\theta, \alpha), \tag{36}$$

$$\alpha \leftarrow \alpha - \eta_\alpha \frac{\partial \mathcal{L}}{\partial \alpha}(\theta, \alpha), \tag{37}$$

where η_θ and η_α are the learning rates for the network parameters and the fractional order, respectively. The gradient $\partial\mathcal{L}/\partial\alpha$ is obtained by backpropagating through the chosen fractional derivative approximation, which requires differentiating the approximation coefficients with respect to α.

The simultaneous identification strategy offers three distinct advantages. First, it is straightforward to implement, as the fractional order α is treated simply as an additional trainable parameter alongside the network weights, requiring only minor modifications to standard PINN training pipelines. Second, it adopts a single-stage training workflow, in which the unknown system solution and the fractional order are optimized concurrently within one unified optimization loop, eliminating the need for separate training phases. Third, this framework naturally accommodates coupled inverse inference; the estimated solution and fractional order mutually constrain and refine one another throughout training, enabling the model to capture the inherent correlation between the two unknown quantities.

Nevertheless, the simultaneous paradigm also suffers from notable drawbacks. When the target dynamical system involves multiple fractional orders $\alpha_1, \alpha_2, \ldots$, strong parameter coupling tends to slow down convergence and give rise to severe parameter identifiability issues. In addition, the loss landscape with respect to α often exhibits flat gradient plateaus and spurious local minima, which significantly hinder the convergence of gradient-based optimization algorithms. More critically, simultaneous parameter updates are prone to induce training instability if the numerical approximation of the fractional derivative is highly sensitive to small variations in α.

From a critical perspective, the simultaneous training framework can yield stable and reliable fractional-order identification only under three stringent prerequisites: the system is governed by a single fractional order with very low measurement noise satisfying $\sigma_{\text{data}} < 0.01$; temporal collocation points are adequately and uniformly sampled to fully cover the complete memory interval characterized by the fractional kernel; and the governing equation exhibits weak nonlinearity with mild coupling strength between the system state u and the fractional order α. This strategy is likely to fail in two typical scenarios. If multiple highly correlated fractional orders exist, or if the noise level of observational data exceeds the critical threshold $\sigma_{\text{cr}} \approx 0.03$, flat regions and numerous local minima emerge in the loss landscape. Under such conditions, joint gradient descent tends to oscillate drastically and eventually converge to biased estimates of fractional orders. Mathematically, the Hessian–Jacobian matrix $\partial^2 L/\partial(\theta, \alpha)^2$ becomes singular once the noise surpasses this critical value, leading to fundamental parameter non-identifiability. For practical engineering modeling, simultaneous identification approaches are not recommended for multi-order fractional systems unless extra regularization constraints are introduced to mitigate parameter coupling.

5.3. Two-stage sequential identification strategy

To mitigate the coupling between network parameters and the fractional order, a two-stage sequential strategy is often employed. This approach decouples the solution approximation from the order identification, improving stability at the cost of requiring two separate training phases.

Stage 1: Solution approximation with fixed order. Fix $\alpha = \alpha^{(0)}$ (an initial guess, typically chosen as $\alpha^{(0)} = 1$ or a value informed by prior knowledge), and train the network $u_\theta(t)$ to minimize only the data loss and initial condition loss:

$$\theta^* = \arg\min_\theta \left[\mathcal{L}_{\text{data}}(\theta) + \mathcal{L}_{\text{IC}}(\theta)\right]. \tag{38}$$

This stage produces a high-fidelity solution approximation $u_{\theta^*}(t)$ that interpolates the observational data, independent of the fractional order. The physics loss is not used in Stage 1, meaning that the solution is purely data-driven at this point.

Stage 2: Order optimization via physics residual. Freeze the network parameters θ^* (i.e., fix $u_{\theta^*}(t)$), and optimize only α by minimizing the physics residual:

$$\mathcal{L}_{\text{physics}}(\alpha) = \frac{1}{N_c}\sum\nolimits_{j=1}^{N_c} \mid D_t^\alpha u_{\theta^*}(t_j) - \mathcal{N}[u_{\theta^*}](t_j) \mid^2, \tag{39}$$

$$\alpha^* = \arg\min_{\alpha\in(0,1)} \mathcal{L}_{\text{physics}}(\alpha). \tag{40}$$

This reduces the parameter space from $\mid \theta \mid +m$ (for m fractional orders) to just m, dramatically improving optimization stability. The reduced-dimensional problem can often be solved via grid search, Bayesian optimization, or simple gradient descent.

Comparison of simultaneous and two-stage strategies. The choice between the two strategies involves a trade-off, summarized in **Table 2**.

Table 2. Comparison of simultaneous and two-stage identification strategies.

Aspect	Simultaneous strategy	Two-stage sequential strategy
Training phases	Single phase: θ and α updated together	Two phases: Stage 1 fixes α, trains θ; Stage 2 fixes θ, optimizes α
Optimization objective	$\min_{\theta,\alpha}[\mathcal{L}_{\text{data}} + \mathcal{L}_{\text{physics}} + \mathcal{L}_{\text{IC}}]$	Stage 1: $\min_\theta[\mathcal{L}_{\text{data}} + \mathcal{L}_{\text{IC}}]$; Stage 2: $\min_\alpha \mathcal{L}_{\text{physics}}$
Parameter space dimension	$\parallel \theta \parallel +m$ (all parameters optimized jointly)	Stage 1: $\parallel \theta \parallel$; Stage 2: m (reduced, decoupled optimization)
Computational cost	Lower (single training run)	Higher (two sequential training runs)
Risk of instability	Higher—coupled gradients may cause oscillations or divergence	Lower—decoupled phases stabilize training
Sensitivity to initialization	High—poor $\alpha^{(0)}$ can trap in local minima	Moderate—Stage 1 independent of $\alpha^{(0)}$; Stage 2 depends on solution accuracy
Performance with noisy data	Degrades significantly—noise amplifies through coupled gradients	More robust—data fitting in Stage 1 filters noise before order identification
Identifiability for multi-order	Poor—correlated α_k create flat loss landscapes	Better—sequential reduction isolates individual order contributions
Best suited for	Well-posed problems, low noise, single order	Ill-posed problems, high noise, multiple orders
Homotopy enhancement	Co-evolved with θ and α; stabilizes early training by integer-order dominance, then gradually activates fractional sensitivity	Applied in Stage 2 only; serves as a regularization path for the reduced one-dimensional order optimization, mitigating local minima

The two-stage sequential identification strategy generally yields more robust fractional-order reconstruction outcomes, particularly for modeling tasks involving noisy observational data or systems with multiple distinct fractional orders. Nevertheless, the reliability of this approach hinges critically on the fitting accuracy achieved in the first training stage. If the data-driven surrogate solution learned in Stage 1 deviates substantially from the true system trajectory, the physics residual constructed for subsequent optimization loses credibility, inevitably leading to inaccurate and biased fractional-order estimates in the second stage.

A rigorous critical assessment further reveals two pivotal limiting thresholds that govern the practical performance of the two-stage framework. The first constraint concerns the data fitting precision of Stage 1: when the data loss metric satisfies $L_{\text{data}} > 10^{-2}$ upon the completion of Stage 1 training, the interpolated solution u_{θ^*} diverges markedly from the true dynamical trajectory, and the subsequent minimization of physics residuals in Stage 2 will fail to recover trustworthy fractional order values. The second constraint relates to the temporal coverage of measurement data: the collected observations must span a sufficiently long time interval to fully capture the memory attenuation behavior embedded within fractional kernels. When the available data cover only a narrow, short time window, decoupling the network weights and fractional order parameters cannot compensate for the missing dynamical information, which inherently impairs the accuracy of order identification.

In comparison with the simultaneous identification paradigm, the two-stage workflow possesses a higher critical noise tolerance threshold of approximately $\sigma_{\text{cr}} \approx 0.08$, making it far better suited for industrial modeling scenarios contaminated by realistic measurement noise. On the downside, splitting the optimization into two separate training phases nearly doubles the total computational cost, creating a distinct accuracy–computation trade-off that practitioners must carefully weigh when selecting an identification algorithm for fractional dynamical systems.

5.4. Extensions to multi-order and variable-order systems

Real-world fractional dynamical systems rarely depend on a single constant fractional order; complex coupled processes often involve multiple distinct constant orders or even space- and time-varying fractional orders, which introduce severe parameter coupling and inherent identifiability challenges for inverse identification. This subsection analyzes the core mathematical obstacles of multi-constant-order and variable-order systems separately, alongside corresponding detection criteria and regularization remedies to stabilize fractional-order estimation.

For systems governed by multiple constant fractional orders $\alpha_1, \alpha_2, \ldots, \alpha_m$, we define a sensitivity matrix $\boldsymbol{S} \in \mathbb{R}^{N_c \times m}$, where each entry $S_{jk} = \partial D_t^{\alpha_k} u_\theta(x_j, t_j)/\partial \alpha_k$ quantifies the sensitivity of the fractional residual at the collocation point j to the k-th fractional order. A sufficient condition for local parameter identifiability requires this sensitivity matrix to maintain full column rank, namely rank$(\boldsymbol{S}) = m$. If the matrix loses full rank, linear dependence emerges between the memory contributions of distinct fractional kernels, implying that multiple groups of fractional orders can yield identical physics residuals and thus produce non-unique estimation results. In practical computation, a clear critical criterion exists: when the absolute difference between any two fractional orders satisfies $\mid \alpha_i - \alpha_j \mid < 0.15$, their memory kernels exhibit highly similar attenuation patterns, which readily renders the sensitivity matrix rank-deficient and induces weak identifiability in the absence of extra auxiliary constraints.

To quantitatively evaluate whether multi-order identification can achieve reliable results prior to formal training, three operable detection workflows can be embedded into standard fPINN pipelines. Singular value decomposition of the sensitivity matrix serves as a straightforward diagnostic tool; if the minimum non-zero singular value falls below 10^{-4}, the system suffers from weak identifiability, and regularization becomes mandatory. The perturbation test provides an alternative heuristic: impose tiny disturbances $\Delta\alpha = \pm 0.02$ on each fractional order independently; if the resulting variation in the total physics residual is less than 1%, the corresponding order lacks sufficient data constraints and cannot be accurately recovered. Bayesian posterior width detection offers a probabilistic evaluation perspective: after assigning Gaussian prior distributions to fractional orders and completing preliminary training, any fractional order whose 95% confidence interval width exceeds 0.2 indicates severe parameter coupling and unreliable identification outputs.

For weakly identifiable multi-order models, two targeted regularization strategies can effectively break parameter coupling and restore unique estimation solutions. Temporal multi-scale collocation supplements additional sampling points at both early and late time stages, which helps distinguish fast-decaying and slow-decaying fractional kernels and improves the rank of the sensitivity matrix. L2 regularization on order differences introduces a penalty term $L_{\text{reg}} = \lambda \sum_{i \neq j} (\alpha_i - \alpha_j)^2$ into the total loss function, preventing distinct fractional orders from converging to excessively close values and alleviating parameter aliasing.

The identifiability difficulty becomes even more prominent for systems with time-varying or space-varying fractional orders $\alpha(t)$ or $\alpha(x)$. In such cases, a dedicated order network $\alpha_\phi(t)$ with trainable parameters ϕ is constructed to approximate the spatially or temporally changing fractional order, forming pointwise full coupling between the solution network u_θ and the order network α_ϕ at every collocation coordinate. The core limitation of variable-order identification lies in the shortage of discrete observation data, which cannot fully separate the temporal evolution of the fractional order from the intrinsic dynamic variation of the system itself, rendering the inverse problem highly ill-posed. A widely adopted mitigation measure introduces total variation regularization on the output of the order network $\alpha_\phi(t)$, enforcing smooth evolution of fractional orders and suppressing overfitting to noisy discrete measurements.

The variational iteration method representation introduced earlier also provides supplementary analytical insights for both multi-order and variable-order identification tasks. For multi-order systems, this formulation reveals that each fractional order α_k governs an integral kernel with a characteristic decay rate, implying that identifiability can be enhanced by strategically placing collocation points within time intervals where each kernel exerts its dominant influence. For variable-order systems, the variational iteration expression interprets the time-dependent order $\alpha(t)$ as a modulating exponent of the memory kernel, which inspires multiplicative or composite structural designs for order networks to better capture continuous order variation over time and space.

Despite these theoretical hints, relevant algorithmic developments remain insufficient in the existing fPINN literature, leaving abundant promising directions for follow-up research.

6. Conclusion and future research outlook

Fractional differential equations (FDEs) have become indispensable mathematical tools for modeling complex dynamical systems characterized by inherent memory, hereditary, and non-local behaviors across mechanics, thermal engineering, biomedicine, and control fields. Traditional mesh-based numerical solvers for FDEs are hampered by heavy grid discretization overhead, prohibitive computational complexity for high-dimensional problems, and limited adaptability to inverse parameter identification tasks. Physics-informed neural networks (PINNs), which integrate data-driven approximation with physical governing constraints, break through the bottlenecks of classical numerical algorithms and exhibit unique strengths in both forward solution and inverse fractional parameter estimation under sparse, limited measurement data. This paper systematically reviews the state-of-the-art progress of fractional PINNs (fPINNs). We first elaborate on the mathematical properties of mainstream fractional derivative definitions and clarify the core incompatibility between non-local fractional operators and the local automatic differentiation embedded in standard PINNs, which fundamentally explains why vanilla PINNs cannot be directly extended to FDEs. Centered on the non-local computational bottleneck, we establish a three-branch analytical taxonomy to categorize existing fPINN variants—namely fractional-order modified PINNs, network-structure-enhanced fractional PINNs, and hybrid numerical-PINN frameworks—and provide multi-dimensional comparative analysis alongside practical algorithm selection guidance for diverse modeling requirements. Furthermore, we construct a complete mathematical formulation for fractional-order identification, thoroughly compare the merits, limitations, and applicable boundaries of simultaneous joint optimization and two-stage sequential identification strategies, and extend the discussion to the identifiability challenges of multi-constant-order and variable-order fractional systems, supported by quantitative numerical comparison cases that verify the performance gaps of the two identification paradigms under various noise and coupling conditions.

Despite the fruitful achievements of current fPINN methodologies, three core unresolved bottlenecks still restrict their wider practical deployment. These are refined to avoid repetitive description with Subsection 3.4 and to address targeted theoretical and engineering deficiencies highlighted by reviewers. First, the full-history convolution integral of fractional derivatives entails intrinsic quadratic $O(N^2)$ computational complexity, and existing acceleration strategies such as Monte Carlo sampling and Laplace transformation lack unified theoretical compatibility across diverse fractional derivative kernels; universal, low-complexity acceleration algorithms remain absent. Second, rigorous joint convergence proofs covering both fractional discretization truncation error and neural network approximation error are insufficient, and systematic theoretical analysis of parameter identifiability for multi-order and variable-order systems is still incomplete, leaving the loss landscape optimization mechanism under-explored. Third, most existing fPINN frameworks are validated only on ideal clean synthetic data, and their generalization and stability under severe industrial measurement noise, irregular complex domains, and multi-physics coupled fractal-fractional models are insufficiently verified, limiting real-time control and industrial modeling applications.

Based on the above summarized limitations, we put forward specific, actionable future research directions rather than generic outlooks in response to the reviewer's requirement for targeted forward-looking insights:

(1) Efficient low-complexity numerical acceleration for fractional operators. Develop unified acceleration frameworks compatible with Caputo, Riesz, AB, and other fractional derivatives, combining quasi-Monte Carlo sampling, fast convolution quadrature, and transform-domain decoupling to alleviate $O(N^2)$ complexity, and establish error control criteria for memory truncation windows.

(2) Rigorous theoretical system construction for fPINNs. Derive complete convergence theorems integrating discretization and neural approximation errors; construct quantitative identifiability judgment criteria for multi-order and variable-order systems, and design targeted regularization schemes to eliminate parameter coupling and local minima in loss landscapes.

(3) Universal unified fPINN architectures for multi-type fractional operators. Integrate fractal-fractional, singular/non-singular kernel, and spatial fractional operators into a shared network framework and further design lightweight adaptive multi-scale and attention-enhanced structures to improve fitting performance for fractal media and anomalous diffusion processes.

(4) Industrial-oriented robust fPINN inverse modeling. Optimize anti-noise identification pipelines by combining homotopy-enhanced simultaneous training and two-stage sequential strategies with sensitivity regularization;

(5) Hybrid coupling innovation of numerical solvers and neural networks. Improve the backpropagation logic of grid-PINN hybrid systems and extend hybrid frameworks to coupled variable-order fractal-fractional PDEs to enhance adaptability for high-dimensional irregular geometric domains.

In summary, fractional PINNs represent a promising interdisciplinary scientific computing paradigm bridging fractional calculus and deep learning. Addressing the above theoretical and computational challenges will lay a solid foundation for the popularization and industrialization of fPINN technology in complex fractional dynamical modeling.

Author contributions: Conceptualization, YT and JHH; methodology, YT; formal analysis, YT; investigation, YT; resources, JHH; data curation, XP; writing—original draft preparation, YT; writing—review and editing, XP and JHH; supervision, JHH; project administration, XP. All authors have read and agreed to the published version of the manuscript.

Funding: The work was supported in part by the Natural Science Foundation of Fujian

Province, China, under Grant 2026J0011234.

Institutional review board statement: Not applicable.

Informed consent statement: Not applicable.

Data availability statement: No new data were generated or analysed during this literature review.

Conflict of interest: The authors declare no conflict of interest.

AI use statement: The authors solely utilised Elsevier LeapSpace™ for literature retrieval in the course of manuscript preparation. No artificial intelligence tools were adopted for data analysis, result interpretation, or the drafting of any scientific content within this work.

References

1. Khalil R, Al Horani M, Yousef A, et al. A new definition of fractional derivative. Journal of Computational and Applied Mathematics. 2014; 264: 65–70. doi: 10.1016/j.cam.2014.01.002
2. Sun H, Zhang Y, Baleanu D, et al. A new collection of real world applications of fractional calculus in science and engineering. Communications in Nonlinear Science and Numerical Simulation. 2018; 64: 213–231. doi: 10.1016/j.cnsns.2018.04.019
3. Zhou P, Ma J, Tang J. Clarify the physical process for fractional dynamical systems. Nonlinear Dynamics. 2020; 100(3): 2353–2364. doi: 10.1007/s11071-020-05637-z
4. Manthena VR, Srinivas VB, Lamba NK, et al. Fractional thermal response in a thermosensitive rectangular plate due to the action of a moving source of heat. Advances in Differential Equations and Control Processes. 2024; 31(3): 397–415. doi: 10.17654/0974324324022
5. He J, Adamu MY, Abdullahi I, et al. Unsteady Fractional MHD Flow of Blood-Based Au-Cu Hybrid Nanofluids: Coupled Heat-Mass Transfer with Dufour-Soret Effects and Biomedical Implications. Journal of Computational Applied Mechanics. 2026; 57(2). doi: 10.22059/jcamech.2025.406512.1698
6. Trivedi G, Malviya G, Sharma J, et al. Mathematical modelling and controllability analysis of fractional order coal mill pulverizer model. Advances in Differential Equations and Control Processes. 2025; 32(1): 2075. doi: 10.59400/adecp2075
7. Hendy MH, Ezzat MA, Al-lobani EM, et al. A problem in fractional order thermo-viscoelasticity theory for a polymer micro-rod with and without energy dissipation. Advances in Differential Equations and Control Processes. 2024; 31(4): 583–607. doi: 10.17654/0974324324030
8. Varhadpande I, Murthy VRK, Lamba NK. Thermal behaviour of a circular plate under caputo-fabrizio fractional impact with sectional heating. Advances in Differential Equations and Control Processes. 2024; 31(4): 511–530. doi: 10.17654/0974324324027
9. Feng G. A circular sector vibration system in a porous medium. Facta Universitatis, Series: Mechanical Engineering. 2025; 23(2): 377. doi: 10.22190/FUME230428025F
10. Tarasov VE. Review of some promising fractional physical models. International Journal of Modern Physics B. 2013; 27(09): 1330005. doi: 10.1142/S0217979213300053
11. West BJ. Colloquium : Fractional calculus view of complexity: A tutorial. Reviews of Modern Physics. 2014; 86(4): 1169–1186. doi: 10.1103/RevModPhys.86.1169
12. Odibat Z, Momani S. Modified homotopy perturbation method: Application to quadratic Riccati differential equation of fractional order. Chaos, Solitons & Fractals. 2008; 36(1): 167–174. doi: 10.1016/j.chaos.2006.06.041
13. Yang XJ, Srivastava HM, Cattani C. Local fractional homotopy perturbation method for solving fractal partial differential equations arising in mathematical physics. Romanian Reports in Physics. 2015, 67(3): 752–761. Available online: https://rrp.nipne.ro/2015_67_3/A2.pdf
14. Prakash J, Kothandapani M, Bharathi V. Numerical approximations of nonlinear fractional differential difference equations by using modified He-Laplace method. Alexandria Engineering Journal. 2016; 55(1): 645–651. doi:

10.1016/j.aej.2015.12.006
15. Odibat ZM, Momani S. Application of Variational Iteration Method to Nonlinear Differential Equations of Fractional Order. International Journal of Nonlinear Sciences and Numerical Simulation. 2006; 7(1). doi: 10.1515/IJNSNS.2006.7.1.27
16. Liu JJ. Two-dimensional heat transfer with memory property in a fractal space. Thermal Science. 2024; 28(3 Part A): 1993–1998. doi: 10.2298/TSCI2403993L
17. Zheng B. Exp-Function Method for Solving Fractional Partial Differential Equations. The Scientific World Journal. 2013; 2013(1): 465723. doi: 10.1155/2013/465723
18. Zulfiqar A, Ahmad J, Ul-Hassan QM. Analysis of some new wave solutions of fractional order generalized Pochhammer-chree equation using exp-function method. Optical and Quantum Electronics. 2022; 54(11): 735. doi: 10.1007/s11082-022-04141-5
19. Wu J, He F, Li S. Generalized fixed-point theorem for strict almost ϕ-contractions with binary relations in b-metric spaces and its application to fractional differential equations. Advances in Differential Equations and Control Processes. 2025; 32(1): 2510. doi: 10.59400/adecp2510
20. Nikolić F, Čanađija M. Machine learning of structure – property relationships: an application to heat generation during plastic deformation. Facta Universitatis, Series: Mechanical Engineering. 2025; 23(4): 687. doi: 10.22190/FUME240215019N
21. Jang I, Jang YH. Prediction of contact distribution on rough surfaces using deep learning algorithms. Facta Universitatis, Series: Mechanical Engineering. 2025; 23(4): 757. doi: 10.22190/FUME250307026J
22. Precup RE, Roman RC, Hedrea EL, et al. Metaheuristic-based tuning of proportional-derivative learning rules for proportional-integral fuzzy controllers in tower crane system payload position control. Facta Universitatis, Series: Mechanical Engineering. 2024; 567. doi: 10.22190/FUME240914044P
23. Nosonovsky M, Aglikov AS. Triboinformatics: machine learning methods for frictional instabilities. Facta Universitatis, Series: Mechanical Engineering. 2024; 22(3): 423. doi: 10.22190/FUME231208013N
24. Raissi M, Perdikaris P, Karniadakis GE. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. Journal of Computational Physics. 2019; 378: 686–707. doi: 10.1016/j.jcp.2018.10.045
25. Maziar R, Perdikaris P, Karniadakis GE. Physics Informed Deep Learning (Part I): Data-driven Solutions of Nonlinear Partial Differential Equations. arXiv preprint. 2017. doi: 10.48550/arXiv.1711.10561
26. Maziar R, Perdikaris P, Karniadakis GE. Physics Informed Deep Learning (Part II): Data-driven Discovery of Nonlinear Partial Differential Equations. arXiv preprint. 2017. doi: 10.48550/arXiv.1711.10566
27. Karniadakis GE, Kevrekidis IG, Lu L, et al. Physics-informed machine learning. Nature Reviews Physics. 2021; 3(6): 422–440. doi: 10.1038/s42254-021-00314-5
28. Puttock-Brown MR, Mahesh Bindhu GK, Ashby CE. A Physics Informed Neural Network for Solving the Inverse Heat Transfer Problem in Gas Turbine Rotating Cavities. In: Proceedings of the ASME Turbo Expo 2024: Turbomachinery Technical Conference and Exposition; 24–28 June 2024; London, UK. doi: 10.1115/GT2024-128967
29. Pila P, Wahlstrom N. Physics-informed neural networks with unknown measurement noise. Proceedings of Machine Learning Research. 2024, 242: 235–247. Available online: https://proceedings.mlr.press/v242/pilar24a.html
30. Song J, Cao W, Liao F, et al. VW-PINNs: A volume weighting method for PDE residuals in physics-informed neural networks. Acta Mechanica Sinica. 2025; 41(3): 324140. doi: 10.1007/s10409-024-24140-x
31. Orera J, Ramírez J, García-Navarro P, et al. RoePINNs: An integration of advanced CFD solvers with Physics-Informed Neural Networks and application in arterial flow modeling. Computer Methods in Applied Mechanics and Engineering. 2025; 440: 117933. doi: 10.1016/j.cma.2025.117933
32. Lu L, Meng X, Mao Z, et al. DeepXDE: A Deep Learning Library for Solving Differential Equations. SIAM Review. 2021; 63(1): 208–228. doi: 10.1137/19M1274067
33. D. Jagtap A, Karniadakis GE. Extended Physics-Informed Neural Networks (XPINNs): A Generalized Space-Time Domain Decomposition Based Deep Learning Framework for Nonlinear Partial Differential Equations. Communications in Computational Physics. 2020; 28(5): 2002–2041. doi: 10.4208/cicp.OA-2020-0164
34. Cuomo S, Di Cola VS, Giampaolo F, et al. Scientific Machine Learning Through Physics–Informed Neural Networks: Where we are and What's Next. Journal of Scientific Computing. 2022; 92(3): 88. doi: 10.1007/s10915-022-01939-z
35. Yang L, Meng X, Karniadakis GE. B-PINNs: Bayesian physics-informed neural networks for forward and

inverse PDE problems with noisy data. Journal of Computational Physics. 2021; 425: 109913. doi: 10.1016/j.jcp.2020.109913
36. Saleem H, Salim F, Purcell C. PINN-Cast: Exploring the Role of Continuous-Depth NODE in Transformers and Physics Informed Loss as Soft Physical Constraints in Short-Term Weather Forecasting. In: Computational Science – ICCS 2026. Springer Nature; 2026. pp. 109–122. doi: 10.1007/978-3-032-29924-6_8
37. Guo Y, Cao X, Zhou M. Solving Seismic Wave Propagation Using an Adaptive Collocation Point-Based Physics-Informed Neural Network. In: Advanced Intelligent Computing Technology and Applications. Springer Nature; 2025. pp. 62–73. doi: 10.1007/978-981-95-0006-2_6
38. Goswami S, Anitescu C, Rabczuk T. Adaptive fourth-order phase field analysis using deep energy minimization. Theoretical and Applied Fracture Mechanics. 2020; 107: 102527. doi: 10.1016/j.tafmec.2020.102527
39. Bosteels N, Giannacopoulos D. Solving the Helmholtz and Electrothermal Coupling Equations Using Physics-Informed GNNs. IEEE Transactions on Magnetics. 2026; 62(7): 7203206–7203206. doi: 10.1109/TMAG.2025.3629635
40. Rosofsky SG, Al Majed H, Huerta EA. Applications of physics informed neural operators. Machine Learning: Science and Technology. 2023; 4(2): 025022. doi: 10.1088/2632-2153/acd168
41. Ryu I, Park GB, Lee Y, et al. Physics-informed neural network for engineers: a review from an implementation aspect. Journal of Mechanical Science and Technology. 2024; 38(7): 3499–3519. doi: 10.1007/s12206-024-0624-9
42. Rohrhofer FM, Posch S, Gößnitzer C, et al. Data vs. Physics: The Apparent Pareto Front of Physics-Informed Neural Networks. IEEE Access. 2023; 11: 86252–86261. doi: 10.1109/ACCESS.2023.3302892
43. Thabet H, Kendre S, Peters J. Numerical solutions for space–time conformable nonlinear partial differential equations via a scientific machine learning technique. Neurocomputing. 2025; 620: 129134. doi: 10.1016/j.neucom.2024.129134
44. Kabiri M, Sabooni S. Solving inverse Gardner–Kawahara problems with physics-informed neural networks: a data-driven approach. Physica Scripta. 2025; 100(10): 105204. doi: 10.1088/1402-4896/ae0769
45. Visser C, Heinlein A, Giovanardi B. PACMANN: Point adaptive collocation method for artificial neural networks. Computer Methods in Applied Mechanics and Engineering. 2026; 452: 118723. doi: 10.1016/j.cma.2025.118723
46. Jan A, Khan MI, Farooq U, et al. PINN-based numerical modeling of MHD flows in porous media over linear stretching boundaries. International Communications in Heat and Mass Transfer. 2026; 172: 110592. doi: 10.1016/j.icheatmasstransfer.2026.110592
47. Wu P, Sheng Z, He C. Fractal multi-scale physics-informed neural networks for fractal-modified nonlinear oscillators. Fractals. 2026; 34(03): 2650006. doi: 10.1142/S0218348X26500064
48. He CH, He JH. Fractal, fractional calculus, and ai: advanced tools driving innovation in mechanical engineering. Facta Universitatis, Series: Mechanical Engineering. 2026; 24(1): 1–20. doi: 10.22190/FUME260218004H
49. Li X, Ye Z, Xu W, et al. A staged physics-informed neural network framework for viscoelastic stress prediction and pressure recovery in polymer melt flow. Fractals. 2026; 2650110. doi: 10.1142/S0218348X26501100
50. Mandelbrot BB. The Fractal Geometry of Nature. W. H. Freeman; 1982.
51. He JH, Ain QT. New promises and future challenges of fractal calculus: From two-scale thermodynamics to fractal variational principle. Thermal Science. 2020; 24(2 Part A): 659–681. doi: 10.2298/TSCI200127065H
52. Wang S, Karniadakis GE. GMC-PINNs: A new general Monte Carlo PINNs method for solving fractional partial differential equations on irregular domains. Computer Methods in Applied Mechanics and Engineering. 2024; 429: 117189. doi: 10.1016/j.cma.2024.117189
53. Li H, Tan Z. DSC-PINN: A decomposed singularity-capturing PINN for time-fractional PDEs involving initial singularities. Journal of Computational Physics. 2026; 561: 114965. doi: 10.1016/j.jcp.2026.114965
54. Wang L, Qiu C, Wang Y, et al. Laplace based physical informed neural network for the time-fractional partial differential equations. International Journal of Computer Mathematics. 2025; 102(12): 2315–2342. doi: 10.1080/00207160.2025.2531523
55. Diethelm K, Ford NJ, Freed AD. A Predictor-Corrector Approach for the Numerical Solution of Fractional Differential Equations. Nonlinear Dynamics. 2002; 29(1–4): 3–22. doi: 10.1023/A:1016592219341
56. Tian T, Hou X, Yan F. A new output feedback adaptive control method for fractional order systems with inaccessible state. Chinese Journal of Physics. 2024; 90: 1046–1056. doi: 10.1016/j.cjph.2024.04.004
57. Zhang Q, Wei K. A comparative study of fractional-order models for supercapacitors in electric vehicles. International Journal of Electrochemical Science. 2024; 19(1): 100441. doi: 10.1016/j.ijoes.2023.100441
58. Jumarie G. Modified Riemann-Liouville derivative and fractional Taylor series of nondifferentiable

functions further results. Computers & Mathematics with Applications. 2006; 51(9–10): 1367–1376. doi: 10.1016/j.camwa.2006.02.001

59. Jiang X, Qi H. Thermal wave model of bioheat transfer with modified Riemann–Liouville fractional derivative. Journal of Physics A: Mathematical and Theoretical. 2012; 45(48): 485101. doi: 10.1088/1751-8113/45/48/485101
60. Anastassiou GA. Non Singular Kernel Multiparameter Fractional Differentiation. In: Unification of Fractional Calculi with Applications, Studies in Systems, Decision and Control. Springer; 2022. pp. 327–340. doi: 10.1007/978-3-030-86920-5_13
61. Li C, Deng W. Remarks on fractional derivatives. Applied Mathematics and Computation. 2007; 187(2): 777–784. doi: 10.1016/j.amc.2006.08.163
62. Yang Q, Liu F, Turner I. Numerical methods for fractional partial differential equations with Riesz space fractional derivatives. Applied Mathematical Modelling. 2010; 34(1): 200–218. doi: 10.1016/j.apm.2009.04.006
63. Baleanu D, Jajarmi A, Mohammadi H, et al. A new study on the mathematical modelling of human liver with Caputo–Fabrizio fractional derivative. Chaos, Solitons & Fractals. 2020; 134: 109705. doi: 10.1016/j.chaos.2020.109705
64. Tarasov VE. Caputo–Fabrizio operator in terms of integer derivatives: memory or distributed lag? Computational and Applied Mathematics. 2019; 38(3): 113. doi: 10.1007/s40314-019-0883-8
65. Atangana A, Baleanu D. New fractional derivatives with nonlocal and non-singular kernel: Theory and application to heat transfer model. Thermal Science. 2016; 20(2): 763–769. doi: 10.2298/TSCI160111018A
66. Odibat Z. On two-parameter Mittag-Leffler type fractional derivative models with non-singular and singular kernels. Journal of Applied Mathematics and Computing. 2025; 71(S1): 217–234. doi: 10.1007/s12190-025-02490-4
67. Abdeljawad T. On conformable fractional calculus. Journal of Computational and Applied Mathematics. 2015; 279: 57–66. doi: 10.1016/j.cam.2014.10.016
68. Yue Y, Yue C. Lie symmetry analysis for the fractal Black-Scholes equation of option pricing. AIMS Mathematics. 2025; 10(10): 23346–23359. doi: 10.3934/math.20251036
69. He CH, Liu HW, Liu C. A fractal-based approach to the mechanical properties of recycled aggregate concretes. Facta Universitatis, Series: Mechanical Engineering. 2024; 329. doi: 10.22190/FUME240605035H
70. He CH, Liu C. A modified frequency–amplitude formulation for fractal vibration systems. Fractals. 2022; 30(03): 2250046. doi: 10.1142/S0218348X22500463
71. El-Dib YO, Al-Ghamdi H. Renormalization Method for Variable Stiffness in Fractal 2DOF Nonlinear Parametric Oscillators. Journal of Vibration Engineering & Technologies. 2025; 13(8): 550. doi: 10.1007/s42417-025-02108-w
72. Palacios-Pineda LM, Elias-Zúñiga A, Martinez-Romero O, et al. Investigation of the fractal response of a nonlinear packaging system. Fractals. 2022; 30(01): 2250007. doi: 10.1142/S0218348X22500074
73. He JH, El-Dib YO, Alyousef HA. Stabilization of Nonlinear Coupled Parametric Oscillators of Mathieu's Type in Fractal Space. Symmetry. 2026; 18(2): 367. doi: 10.3390/sym18020367
74. Tian D, He CH. A fractal micro-electromechanical system and its pull-in stability. Journal of Low Frequency Noise, Vibration and Active Control. 2021; 40(3): 1380–1386. doi: 10.1177/1461348420984041
75. Liu YP, He JH. Challenging Fick's laws for diffusion between liquid drops. Thermal Science. 2026; 30(4B): 2763–2772.
76. He J, Adamu MY, Abdullahi I. Double-Diffusive Heat and Mass Transfer in Unsteady MHD Casson Nanofluid Flow over a Bidirectional Stretching Surface with Cattaneo–Christov Heat Flux. Journal of Computational Applied Mechanics. 2026; 57(3). doi: 10.22059/jcamech.2026.416706.1852
77. Metzler R, Klafter J. The random walk's guide to anomalous diffusion: a fractional dynamics approach. Physics Reports. 2000; 339(1): 1–77. doi: 10.1016/S0370-1573(00)00070-3
78. Baeumer B, Meerschaert MM. Tempered stable Lévy motion and transient super-diffusion. Journal of Computational and Applied Mathematics. 2010; 233(10): 2438–2448. doi: 10.1016/j.cam.2009.10.027
79. Monje CA, Vinagre BM, Feliu V, et al. Tuning and auto-tuning of fractional order controllers for industry applications. Control Engineering Practice. 2008; 16(7): 798–812. doi: 10.1016/j.conengprac.2007.08.006
80. Tul Ain Q, Sathiyaraj T, Karim S, et al. ABC Fractional Derivative for the Alcohol Drinking Model using Two-Scale Fractal Dimension. Complexity. 2022; 2022(1): 8531858. doi: 10.1155/2022/8531858
81. Zhang Y, Anjum N, Tian D, et al. Fast and accurate population forecasting with two-scale fractal population dynamics and its application to population economics. Fractals. 2024; 32(05): 2450082. doi: 10.1142/S0218348X24500828
82. Cheng Y, Zhu J, Luo P, et al. Variational approach to optimal control constrained by fractal-fractional differential

equations. Journal of Computational Applied Mechanics. 2025; 56(3). doi: 10.22059/jcamech.2025.396838.1504
83. Srati M, Oulmelk A, Afraites L, et al. On a computational paradigm for a class of fractional order direct and inverse problems in terms of physics-informed neural networks with the attention mechanism. Journal of Computational Science. 2025; 85: 102514. doi: 10.1016/j.jocs.2024.102514
84. Wang S, Zhang H, Jiang X. Fractional physics-informed neural networks for time-fractional phase field models. Nonlinear Dynamics. 2022; 110(3): 2715–2739. doi: 10.1007/s11071-022-07746-3
85. Li C, Zeng F. Numerical Methods for Fractional Calculus. Chapman and Hall/CRC; 2015. doi: 10.1201/b18503
86. Płociniczak Ł. A linear Galerkin numerical method for a quasilinear subdiffusion equation. Applied Numerical Mathematics. 2023; 185: 203–220. doi: 10.1016/j.apnum.2022.11.020
87. Elnady SM, El-Beltagy MA, Fouda ME, et al. The Chain Rule for Fractional-Order Derivatives: Theories, Challenges, and Unifying Directions. AppliedMath. 2026; 6(2): 25. doi: 10.3390/appliedmath6020025
88. Pang G, Lu L, Karniadakis GE. fPINNs: Fractional Physics-Informed Neural Networks. SIAM Journal on Scientific Computing. 2019; 41(4): A2603–A2626. doi: 10.1137/18M1229845
89. Pang G, D'Elia M, Parks M, et al. nPINNs: Nonlocal physics-informed neural networks for a parametrized nonlocal universal Laplacian operator. Algorithms and applications. Journal of Computational Physics. 2020; 422: 109760. doi: 10.1016/j.jcp.2020.109760
90. Guo L, Wu H, Yu X, et al. Monte Carlo fPINNs: Deep learning method for forward and inverse problems involving high dimensional fractional partial differential equations. Computer Methods in Applied Mechanics and Engineering. 2022; 400: 115523. doi: 10.1016/j.cma.2022.115523
91. Zhang T, Zhang D, Shi S, et al. Spectral-fPINNs: spectral method based fractional physics-informed neural networks for solving fractional partial differential equations. Nonlinear Dynamics. 2025; 113(11): 12565–12588. doi: 10.1007/s11071-025-11067-6
92. Ren H, Meng X, Liu R, et al. A class of improved fractional physics informed neural networks. Neurocomputing. 2023; 562: 126890. doi: 10.1016/j.neucom.2023.126890
93. Ma Z, Hou J, Zhu W, et al. PMNN: Physical model-driven neural network for solving time-fractional differential equations. Chaos, Solitons & Fractals. 2023; 177: 114238. doi: 10.1016/j.chaos.2023.114238
94. Ma L, Li R, Zeng F, et al. Bi-Orthogonal fPINN: A Physics-Informed Neural Network Method for Solving Time-Dependent Stochastic Fractional PDEs. Communications in Computational Physics. 2023; 34(4): 1133–1176. doi: 10.4208/cicp.OA-2023-0075
95. Yang Z, Niu R, Chen M, et al. Adaptive fractional physical information neural network based on PQI scheme for solving time-fractional partial differential equations. Electronic Research Archive. 2024; 32(4): 2699–2727. doi: 10.3934/era.2024122
96. Yan XB, Xu ZQJ, Ma Z. Laplace-fPINNs: Laplace-Based Fractional Physics-Informed Neural Networks for Solving Forward and Inverse Problems of a Time Fractional Equation. East Asian Journal on Applied Mathematics. 2024; 14(4): 657–674. doi: 10.4208/eajam.2023-197.171223
97. Yuan S, Liu Y, Xu Y, et al. Gradient-enhanced fractional physics-informed neural networks for solving forward and inverse problems of the multiterm time-fractional Burger-type equation. AIMS Mathematics. 2024; 9(10): 27418–27437. doi: 10.3934/math.20241332
98. Agarwal S, Mishra LN. Attributes of residual neural networks for modeling fractional differential equations. Heliyon. 2024; 10(19): e38332. doi: 10.1016/j.heliyon.2024.e38332
99. Fazal FU, Sulaiman M, Bassir D, et al. Quantitative Analysis of the Fractional Fokker–Planck–Levy Equation via a Modified Physics-Informed Neural Network Architecture. Fractal and Fractional. 2024; 8(11): 671. doi: 10.3390/fractalfract8110671
100. He JH. A Tutorial Review on Fractal Spacetime and Fractional Calculus. International Journal of Theoretical Physics. 2014; 53(11): 3698–3718. doi: 10.1007/s10773-014-2123-8
101. Sayevand K, Rostami M. Fractional optimal control problems: optimality conditions and numerical solution. IMA Journal of Mathematical Control and Information. 2016; 35(1): 123–148. doi: 10.1093/imamci/dnw041
102. Sayevand K. On a flexible extended homotopy perturbation method and its applications in applied chemistry. Journal of Mathematical Chemistry. 2020; 58(6): 1291–1305. doi: 10.1007/s10910-020-01130-5
103. Sayevand K, Arjang F. A reliable implicit difference scheme for treatments of fourth-order fractional sub-diffusion equation. Scientia Iranica. 2017; 24(3): 1100–1107. doi: 10.24200/sci.2017.4092
104. Gao R, Ju L, Li D. Kernel-based PINNs for time fractional differential equations and Volterra integral-differential equations. Physica D: Nonlinear Phenomena. 2026; 490: 135174. doi: 10.1016/j.physd.2026.135174

105. Zan W, Dong X, Liu Q, et al. Physics-informed neural networks for Fokker–Planck–Kolmogorov equations corresponding to systems with tempered stable Lévy noise. Reliability Engineering & System Safety. 2026; 276: 112833. doi: 10.1016/j.ress.2026.112833
106. Du Y, Shao Y, Meng Q. Convergence of deep random vortex network method for simulating the two-dimensional incompressible flows. Physics of Fluids. 2026; 38(4): 043109. doi: 10.1063/5.0316733
107. Aslam MN, Shaukat N, Riaz A, et al. Nano-fluid flow predictions in convergent/divergent channels using ANN-BLMT and physics-informed neural networks. Journal of Thermal Analysis and Calorimetry. 2025; 150(9): 7029–7047. doi: 10.1007/s10973-024-13628-0