Journal of Computational Mathematics and Physics
www.jcmp.org

Article Type: Original Research Article

An Uncertainty-Aware Physics-Informed Neural Network Framework for Nonlinear Fractional Partial Differential Equations in Mathematical Physics

Masood Rauf khan[1,*]

[1]National Research Council Institute for SuPerconductors, INnovative Materials, and Devices, CNR-SPIN, University of Salerno, Via Giovanni Paolo II, 132, 84084 Fisciano, Italy.

[*]Corresponding Author: masoodrauf.khan@spin.cnr.it

Received: 12 April 2026 **Accepted:** 19 July 2026 **Published:** 25 July 2026

Abstract

Fractional partial differential equations (FPDEs) provide effective mathematical descriptions of memory, hereditary behavior, nonlocal transport, anomalous diffusion, and multiscale dynamics. Their numerical solution, however, remains expensive because fractional operators are nonlocal and because nonlinear models frequently involve uncertain coefficients, incomplete observations, and irregular solution regularity. This study develops a unified uncertainty-aware fractional physics-informed neural network (UA-fPINN) framework for forward and inverse nonlinear FPDEs. The proposed methodology combines: (i) a neural representation of the state variable; (ii) a consistent quadrature approximation of the Caputo time derivative; (iii) automatic differentiation for local spatial derivatives; (iv) adaptive balancing of initial, boundary, observation, and physics residual losses; and (v) probabilistic uncertainty quantification through deep ensembles or Bayesian weight inference. A coercivity-based residual-to-solution error estimate is established under standard monotonicity and consistency assumptions. The analysis separates neural approximation, quadrature, sampling, optimization, and observational errors, thereby clarifying when a small training loss is expected to imply a small physical solution error. A manufactured-solution benchmark for a nonlinear time-fractional reaction–diffusion equation is formulated, together with a reproducible experimental protocol, sensitivity measures, calibration diagnostics, and comparison criteria against the L1 finite-difference method and deterministic fPINNs. The framework is designed for applications in anomalous transport, viscoelasticity, porous media, thermal processes, biological systems, and inverse parameter identification. The study provides a mathematically transparent basis for reliable scientific machine learning in nonlocal mathematical physics.

Keywords: Fractional partial differential equations, Physics-informed neural networks, Caputo derivative, Inverse problems, error analysis; anomalous diffusion; scientific machine learning.

1. Introduction

Mathematical physics increasingly requires models that combine nonlinearity, nonlocality, memory, uncertain parameters, and incomplete observations. Contemporary problems range from anomalous diffusion and viscoelastic relaxation to wave propagation, fluid transport, biological dynamics, inverse scattering, and ill-posed boundary-value problems. A broad overview of these themes is provided by [**?**], who highlighted fractional equations, differential and integral equations, ill-posed problems, numerical analysis, fuzzy uncertainty, and mathematical modeling as active directions in modern mathematical physics.

Classical integer-order partial differential equations often assume that the instantaneous rate of change depends only on the current state. This Markovian assumption is inadequate for many materials and transport processes that retain a history of earlier states. Fractional calculus introduces derivatives and integrals of noninteger order and therefore provides compact representations of memory and long-range interactions. Foundational treatments by [20], [13], and [5] established the analytical basis of fractional differential equations, while [17] demonstrated their relevance to anomalous diffusion and random-walk models.

The numerical solution of FPDEs is substantially more demanding than that of their integer-order counterparts. A Caputo derivative at time t depends on the entire temporal history over $[0, t]$, so direct time stepping typically has increasing memory and computational costs. Classical approaches include convolution quadrature [16], predictor–corrector methods [4], finite-difference/spectral methods [30], graded temporal meshes [23], and corrected L1-type formulas [10, 6]. These methods are reliable when a computational mesh is available and the model coefficients are known, but repeated solution of high-dimensional forward problems or inverse estimation can be expensive.

Physics-informed neural networks (PINNs) offer a mesh-free scientific machine-learning alternative. In the PINN methodology, a neural network approximates the solution and is trained by minimizing the residual of the governing equation together with initial, boundary, and observation mismatches [**?**] The method has attracted significant attention because it incorporates known physical laws into learning and can address both forward and inverse problems [12, 2]. Fractional PINNs (fPINNs) were introduced by [19], who combined automatic differentiation for integer-order operators with numerical discretization of fractional operators.

Despite this progress, three limitations remain central. First, the fractional derivative is not handled exactly by automatic differentiation; its discretization error becomes part of the training process. Second, PINN optimization may suffer from gradient imbalance, spectral bias, stiffness, and poorly scaled loss terms [24, 25, 14]. Third, deterministic predictions provide no direct measure of confidence when observations, parameters, or neural approximations are uncertain. Bayesian PINNs and probabilistic physics-informed models address parts of this problem [27, 29, 26], but uncertainty-aware formulations for nonlinear FPDEs require additional analysis because discretization, nonlocality, and training uncertainty interact.

The present article develops a unified uncertainty-aware fPINN framework. Its main contributions are:

1. a general formulation for nonlinear time-fractional PDEs with uncertain coefficients and noisy data;
2. a hybrid computational operator using an L1-type approximation for the Caputo derivative and automatic differentiation for local spatial terms;
3. adaptive loss weighting and residual-based collocation refinement;
4. ensemble and Bayesian strategies for epistemic and aleatoric uncertainty;
5. a residual-to-solution error estimate that explicitly separates quadrature, approximation, sampling, optimization, and data errors; and

6. a reproducible manufactured-solution benchmark and evaluation protocol that avoids unsupported empirical claims.

2. Mathematical Background and Related Work

2.1. Caputo fractional derivative

For $0 < \alpha < 1$, the Caputo derivative of an absolutely continuous function v is

$$ {}^{C}D_0^{\alpha} v(t) = \frac{1}{\Gamma(1-\alpha)} \int_0^t (t-s)^{-\alpha} v'(s)\, ds. \tag{1} $$

The Caputo form is particularly useful in physical models because it permits classical initial conditions such as $v(0) = v_0$ [1, 20]. The weakly singular kernel gives high weight to recent history while retaining the influence of earlier states.

A useful identity for $\beta > -1$ is

$$ {}^{C}D_0^{\alpha} t^{\beta} = \frac{\Gamma(\beta+1)}{\Gamma(\beta+1-\alpha)} t^{\beta-\alpha}. \tag{2} $$

This identity will be used to construct the manufactured solution in

2.2. L1 approximation

Let $0 = t_0 < t_1 < \cdots < t_M = T$, with uniform step $\tau = T/M$. The L1 approximation to (1) at t_n is

$$ \delta_{\tau}^{\alpha} v^n = \frac{\tau^{-\alpha}}{\Gamma(2-\alpha)} \sum_{k=0}^{n-1} a_k \left(v^{n-k} - v^{n-k-1} \right), \qquad a_k = (k+1)^{1-\alpha} - k^{1-\alpha}. \tag{3} $$

For sufficiently smooth v, the local consistency error is of order $O(\tau^{2-\alpha})$. Fractional solutions often have an initial weak singularity, and the actual order can deteriorate unless graded meshes or correction terms are used [10, 23]. This issue is important for fPINNs because the fractional residual cannot be more accurate than the underlying quadrature.

2.3. Physics-informed neural networks

Let $u_\theta(\boldsymbol{x}, t)$ be a neural network with trainable parameters θ. For a PDE

$$ \mathcal{F}[u](\boldsymbol{x}, t) = 0, $$

a standard PINN minimizes a weighted sum

$$ \mathcal{J}(\theta) = \lambda_r \mathcal{J}_r(\theta) + \lambda_b \mathcal{J}_b(\theta) + \lambda_0 \mathcal{J}_0(\theta) + \lambda_d \mathcal{J}_d(\theta), \tag{4} $$

where the terms represent interior residual, boundary, initial, and data losses. Spatial and integer-order temporal derivatives are obtained by automatic differentiation. The approach is mesh-free in the local differential operators, but fractional operators require a history-aware numerical rule [19].

PINNs can fail even when the network is expressive. Unequal gradient magnitudes across loss components can cause the optimizer to satisfy one part of the problem while neglecting another [24]. Neural tangent kernel analysis further indicates that different components of the target solution may be learned at different rates [25]. Curriculum training, adaptive activations, nondimensionalization, domain decomposition, and residual-based sampling are therefore important practical elements [8, 9].

2.4. Uncertainty in scientific machine learning

Uncertainty can be divided into aleatoric and epistemic components. Aleatoric uncertainty represents irreducible randomness or measurement noise. Epistemic uncertainty represents limited knowledge, sparse data, uncertain coefficients, and uncertainty in the learned model. Bayesian PINNs assign probability distributions to network weights or physical parameters [27]. Alternative approaches use polynomial chaos, dropout, deep ensembles, or adversarial generative models [29, 26]. For FPDEs, a complete uncertainty budget should also include the deterministic error introduced by fractional quadrature.

3. Problem Formulation

Let $\Omega \subset R^d$ be a bounded Lipschitz domain, $T > 0$, and $Q = \Omega \times (0, T]$. We consider

$$
\begin{aligned}
{}^{C}D_0^{\alpha} u(\boldsymbol{x}, t, \boldsymbol{\xi}) + \mathcal{L}_{\boldsymbol{\xi}} u(\boldsymbol{x}, t, \boldsymbol{\xi}) + \mathcal{N}(u(\boldsymbol{x}, t, \boldsymbol{\xi}); \boldsymbol{\xi}) &= f(\boldsymbol{x}, t, \boldsymbol{\xi}), && (\boldsymbol{x}, t) \in Q, && (5)\\
u(\boldsymbol{x}, 0, \boldsymbol{\xi}) &= u_0(\boldsymbol{x}, \boldsymbol{\xi}), && \boldsymbol{x} \in \Omega, && (6)\\
\mathcal{B} u(\boldsymbol{x}, t, \boldsymbol{\xi}) &= g(\boldsymbol{x}, t, \boldsymbol{\xi}), && (\boldsymbol{x}, t) \in \partial\Omega \times (0, T]. && (7)
\end{aligned}
$$

Here $0 < \alpha < 1$, $\boldsymbol{\xi}$ denotes uncertain parameters, $\mathcal{L}_{\boldsymbol{\xi}}$ is a possibly parameter-dependent elliptic operator, $\mathcal{N}$ is nonlinear, and $\mathcal{B}$ is a boundary operator.

A representative reaction–diffusion operator is

$$
\mathcal{L}_{\boldsymbol{\xi}} u = -\nabla \cdot (\kappa(\boldsymbol{x}, \boldsymbol{\xi}) \nabla u), \qquad \mathcal{N}(u; \boldsymbol{\xi}) = \rho(\boldsymbol{\xi}) u^3. \tag{8}
$$

The unknowns may include the field u, the fractional order α, diffusivity κ, reaction coefficient ρ, or selected source parameters.

Noisy observations are modeled by

$$
y_i = u(\boldsymbol{x}_i, t_i, \boldsymbol{\xi}_i) + \varepsilon_i, \qquad \varepsilon_i \sim \mathcal{N}(0, \sigma_i^2), \quad i = 1, \ldots, N_d. \tag{9}
$$

Assumption (Well-posedness and Monotonicity)

For almost every realization $\boldsymbol{\xi}$, the operator $\mathcal{L}_{\boldsymbol{\xi}}$ is coercive on a Hilbert space $V \subset L^2(\Omega)$, and $\mathcal{N}$ is locally Lipschitz and monotone on the relevant bounded solution set. The data are compatible and the problem admits a unique weak solution.

These assumptions are consistent with standard existence and uniqueness frameworks for fractional evolution problems [3, 5]. More singular nonlinearities or nonmonotone dynamics require problem-specific arguments.

4. Uncertainty-Aware Fractional PINN

4.1. Neural representation

We approximate the solution by

$$
u_\theta(\boldsymbol{x}, t, \boldsymbol{\xi}) = u_{\text{lift}}(\boldsymbol{x}, t, \boldsymbol{\xi}) + \phi(\boldsymbol{x}, t) N_\theta(\boldsymbol{x}, t, \boldsymbol{\xi}), \tag{10}
$$

where N_θ is a fully connected neural network. The lifting function satisfies the prescribed initial and, when convenient, boundary conditions. The vanishing factor ϕ enforces homogeneous constraints exactly. Exact constraint embedding reduces competition between loss components and can improve training stability.

For uncertain inputs, $\boldsymbol{\xi}$ is appended to the network input. The network therefore learns the parametric map

$$
(\boldsymbol{x}, t, \boldsymbol{\xi}) \mapsto u(\boldsymbol{x}, t, \boldsymbol{\xi}),
$$

rather than solving a new deterministic problem for every realization.

4.2. Hybrid fractional residual

At the temporal grid $\{t_n\}_{n=0}^{M}$, the fractional derivative is approximated by

$$\widehat{{}^{C}D_{0}^{\alpha}}u_\theta(\boldsymbol{x},t_n,\boldsymbol{\xi}) = \frac{\tau^{-\alpha}}{\Gamma(2-\alpha)}\sum_{k=0}^{n-1} a_k\left[u_\theta(\boldsymbol{x},t_{n-k},\boldsymbol{\xi}) - u_\theta(\boldsymbol{x},t_{n-k-1},\boldsymbol{\xi})\right]. \tag{11}$$

Spatial derivatives are computed by automatic differentiation. The residual is

$$r_\theta(\boldsymbol{x},t_n,\boldsymbol{\xi}) = \widehat{{}^{C}D_{0}^{\alpha}}u_\theta + \mathcal{L}_{\boldsymbol{\xi}}u_\theta + \mathcal{N}(u_\theta;\boldsymbol{\xi}) - f. \tag{12}$$

The method is "mesh-free" only in space and parameter dimensions; the temporal history grid is necessary for the nonlocal derivative. Fast convolution, sum-of-exponentials kernels, or short-memory approximations can reduce the $O(M^2)$ history cost for long-time simulations.

4.3. Composite objective

Let $\mathcal{S}_r, \mathcal{S}_b, \mathcal{S}_0, \mathcal{S}_d$ be residual, boundary, initial, and observation samples. Define

$$\mathcal{J}_r = \frac{1}{N_r}\sum_{z\in\mathcal{S}_r}|r_\theta(z)|^2, \tag{13}$$

$$\mathcal{J}_b = \frac{1}{N_b}\sum_{z\in\mathcal{S}_b}|\mathcal{B}u_\theta(z) - g(z)|^2, \tag{14}$$

$$\mathcal{J}_0 = \frac{1}{N_0}\sum_{z\in\mathcal{S}_0}|u_\theta(z) - u_0(z)|^2, \tag{15}$$

$$\mathcal{J}_d = \frac{1}{N_d}\sum_{i=1}^{N_d}\frac{|u_\theta(\boldsymbol{x}_i,t_i,\boldsymbol{\xi}_i) - y_i|^2}{\sigma_i^2}. \tag{16}$$

Regularization and calibration terms may be added:

$$\mathcal{J}_{\text{tot}} = \sum_{q\in\{r,b,0,d\}}\lambda_q\mathcal{J}_q + \lambda_{\text{reg}}\|\theta\|_2^2 + \lambda_{\text{cal}}\mathcal{J}_{\text{cal}}. \tag{17}$$

4.4. Adaptive loss balancing

Fixed weights may create gradient imbalance. We therefore update the weights by equalizing the exponential moving averages of gradient norms:

$$G_q^{(m)} = \beta G_q^{(m-1)} + (1-\beta)\left\|\nabla_\theta\mathcal{J}_q(\theta^{(m)})\right\|_2, \tag{18}$$

and

$$\lambda_q^{(m+1)} = \frac{\overline{G}^{(m)}}{G_q^{(m)} + \epsilon}, \qquad \overline{G}^{(m)} = \frac{1}{4}\sum_q G_q^{(m)}. \tag{19}$$

Weights are clipped to a prescribed interval to avoid extreme updates. This mechanism is motivated by the gradient pathologies analyzed by [24].

4.5. Residual-based adaptive sampling

After a training stage, the residual is evaluated on a candidate pool. New points are sampled with probability

$$p(z) = \frac{(|r_\theta(z)| + \delta)^\gamma}{\sum_{\tilde{z}}(|r_\theta(\tilde{z})| + \delta)^\gamma}. \tag{20}$$

This concentrates collocation points in regions with unresolved layers, initial singularity, rapid nonlinear response, or poor parameter-space coverage.

4.6. Probabilistic prediction

Two uncertainty implementations are recommended.

Deep ensemble. Train K independently initialized UA-fPINNs. The predictive mean and epistemic variance are

$$\widehat{\mu}(z) = \frac{1}{K}\sum_{k=1}^{K} u_{\theta_k}(z), \tag{21}$$

$$\widehat{\sigma}^2_{\mathrm{epi}}(z) = \frac{1}{K-1}\sum_{k=1}^{K}\left(u_{\theta_k}(z) - \widehat{\mu}(z)\right)^2. \tag{22}$$

If the network also predicts a noise variance $\sigma^2_{\mathrm{ale}}(z)$, then

$$\widehat{\sigma}^2_{\mathrm{tot}}(z) = \widehat{\sigma}^2_{\mathrm{epi}}(z) + \frac{1}{K}\sum_{k=1}^{K}\widehat{\sigma}^2_{\mathrm{ale},k}(z). \tag{23}$$

Bayesian PINN. Assign a prior $p(\theta, \boldsymbol{\xi})$ and infer

$$p(\theta, \boldsymbol{\xi} \mid \mathcal{D}) \propto p(\mathcal{D} \mid \theta, \boldsymbol{\xi}) p(\theta, \boldsymbol{\xi}) \exp\left[-\eta \mathcal{J}_r(\theta, \boldsymbol{\xi})\right]. \tag{24}$$

Hamiltonian Monte Carlo is more accurate but expensive; variational inference is more scalable [27]. Deep ensembles are often a practical default, while Bayesian inference is preferable when posterior parameter distributions are scientifically important.

4.7. Workflow

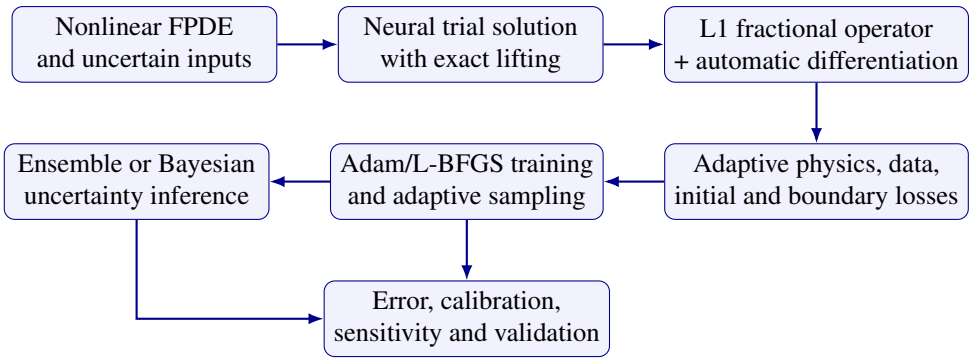

Figure 1: Computational workflow of the proposed UA-fPINN framework.

5. Consistency, Stability, and Error Analysis

A major concern in PINN studies is whether a small empirical loss guarantees a small solution error. The answer depends on operator stability, quadrature consistency, sampling quality, network approximation, and optimization.

assumption(Fractional quadrature consistency)

For a sufficiently regular function v, the discrete operator satisfies

$$\left\| {}^{C}D_0^{\alpha} v - \delta_{\tau}^{\alpha} v \right\|_{L^2(0,T;V')} \le C_{\alpha} \tau^{p}, \tag{25}$$

where $p = 2 - \alpha$ for smooth solutions on a uniform mesh and may be lower for nonsmooth data.

Algorithm 1 Training procedure for UA-fPINN

Require: FPDE, parameter distribution, observations, temporal grid, candidate collocation pool

Construct u_θ using lifting functions
Initialize network parameters and loss weights
for stage $s = 1, \ldots, S$ **do**
    Sample interior, boundary, initial, and parameter points
    Evaluate the L1 history operator and automatic-differentiation terms
    Compute $\mathcal{J}_r, \mathcal{J}_b, \mathcal{J}_0, \mathcal{J}_d$
    Update adaptive weights using (19)
    Optimize with Adam for N_{Adam} iterations
    Refine collocation points using (20)
end for
Fine-tune with L-BFGS
Repeat for K ensemble members or sample a Bayesian posterior
Report predictive mean, uncertainty intervals, residuals, and calibration diagnostics

assumption(Residual stability)

Let u be the exact solution and v an admissible trial function. There exists $C_{\text{stab}} > 0$ such that

$$\|v - u\|_X \le C_{\text{stab}} \left(\|\mathcal{F}(v)\|_Y + \|v(\cdot, 0) - u_0\|_{L^2(\Omega)} + \|\mathcal{B}v - g\|_Z \right), \tag{26}$$

where X is an energy norm and Y, Z are residual and trace spaces.

proposition(Discrete residual-to-error estimate)

the UA-fPINN approximation satisfies

$$\|u_\theta - u\|_X \le C_{\text{stab}} \Big(\|r_{\theta,\tau}\|_Y + \|u_\theta(\cdot, 0) - u_0\|_{L^2(\Omega)} + \|\mathcal{B}u_\theta - g\|_Z + C_\alpha \tau^p \Big), \tag{27}$$

where $r_{\theta,\tau}$ is the residual computed with the discrete fractional operator.

proof

Add and subtract the discrete fractional derivative:

$$\mathcal{F}(u_\theta) = r_{\theta,\tau} + \left({}^{C}D_0^\alpha u_\theta - \delta_\tau^\alpha u_\theta \right).$$

Apply the triangle inequality , then substitute the result into the stability estimate (26).

The proposition shows that the physics loss alone is insufficient: the temporal quadrature error remains even when the empirical residual is nearly zero.

Empirical-to-population residual bound

Suppose the squared residual class is uniformly bounded by B, and N_r collocation points are sampled independently from a distribution μ. Then, with probability at least $1 - \delta$,

$$\left| \mathbb{E}_\mu \left| r_{\theta,\tau} \right|^2 - \frac{1}{N_r} \sum_{i=1}^{N_r} \left| r_{\theta,\tau}(z_i) \right|^2 \right| \le 2\mathfrak{R}_{N_r}(\mathcal{G}) + B \sqrt{\frac{\log(2/\delta)}{2N_r}}. \tag{28}$$

where $\mathfrak{R}_{N_r}(\mathcal{G})$ denotes the empirical Rademacher complexity of the residual class.

proof

The result follows from a standard uniform concentration inequality for bounded function classes applied to $\mathcal{G} = \{|r_{\theta,\tau}|^2 : \theta \in \Theta\}$.

Related PINN generalization estimates have been developed for broad PDE classes by [18]. The bound indicates that increasing collocation size reduces sampling error, but network complexity and residual regularity also matter.

Combining approximation, optimization, sampling, data, and quadrature terms yields the schematic decomposition

$$\mathcal{E}_{\text{total}} \lesssim \mathcal{E}_{\text{app}} + \mathcal{E}_{\text{opt}} + \mathcal{E}_{\text{samp}} + \mathcal{E}_{\text{quad}} + \mathcal{E}_{\text{data}}. \tag{29}$$

Here:

- $\mathcal{E}_{\text{app}}$ is the best-in-class neural approximation error;
- $\mathcal{E}_{\text{opt}}$ is the gap between the achieved and global training objectives;
- $\mathcal{E}_{\text{samp}}$ is the empirical-to-population residual gap;
- $\mathcal{E}_{\text{quad}}$ is the fractional discretization error; and
- $\mathcal{E}_{\text{data}}$ is the propagated observational and parameter uncertainty.

Remark

A narrow ensemble interval does not guarantee accuracy. If every ensemble member shares the same quadrature bias or model-form error, the ensemble may be confidently wrong. Calibration must therefore be assessed against reference data, manufactured solutions, or independently resolved simulations.

6. Manufactured-Solution Benchmark

We consider the nonlinear time-fractional reaction–diffusion equation

$$ {}^{C}D_0^{\alpha} u(x,t) - \kappa u_{xx}(x,t) + \rho u^3(x,t) = f(x,t), \qquad 0 < x < 1, \quad 0 < t \le 1, \tag{30}$$

$$u(0,t) = u(1,t) = 0, \tag{31}$$

$$u(x,0) = 0. \tag{32}$$

Choose

$$u^{\star}(x,t) = t^{\beta} \sin(\pi x), \qquad \beta > 1. \tag{33}$$

Using (2), the forcing term is

$$f(x,t) = \frac{\Gamma(\beta+1)}{\Gamma(\beta+1-\alpha)} t^{\beta-\alpha} \sin(\pi x) + \kappa \pi^2 t^{\beta} \sin(\pi x) + \rho t^{3\beta} \sin^3(\pi x). \tag{34}$$

The benchmark is exact for any admissible $\alpha, \beta, \kappa, \rho$, making it suitable for both forward and inverse testing.

6.1. Uncertain-input version

Let

$$\alpha \sim \mathcal{U}(0.55, 0.85), \qquad \kappa \sim \mathcal{U}(0.05, 0.20), \qquad \rho \sim \mathcal{U}(0.5, 1.5). \tag{35}$$

The source term is evaluated consistently for each realization. A parametric UA-fPINN takes $(x, t, \alpha, \kappa, \rho)$ as inputs. This design tests whether the model can learn an entire solution family rather than a single realization.

6.2. Inverse problem

For inverse identification, α, κ, ρ are treated as trainable variables with positivity or box constraints:

$$\alpha = 0.5 + 0.4\,\sigma(a), \quad \kappa = \exp(b), \quad \rho = \exp(c), \tag{36}$$

where σ is the logistic function. Noisy observations generated from (33) are used to infer the parameters.

7. Reproducible Numerical Protocol

This section specifies an experimental design rather than reporting invented numerical outcomes. A future computational implementation should publish code, random seeds, hardware information, and raw outputs.

7.1. Recommended settings

Table 1: Recommended baseline configuration.

Component	Baseline choice
Network	6 hidden layers, 64 neurons per layer, $\tanh$ activation
Input normalization	Affine scaling of every input to $[-1, 1]$
Temporal grid	$M = 100, 200, 400$ for refinement analysis
Interior points	$N_r = 20{,}000$, with residual-based refinement
Boundary/initial points	$N_b = N_0 = 1{,}000$
Optimizer	Adam followed by L-BFGS
Adam schedule	10^{-3} initially, then cosine or step decay
Ensemble size	$K = 5$ or 10 independent models
Reference method	L1 finite differences with refined spatial mesh
Precision	Float64 for fractional history accumulation
Reporting	Mean and standard deviation over at least five seeds

7.2. Comparison methods

The following methods should be compared:

1. classical L1 finite difference or spectral discretization;
2. deterministic fPINN with fixed loss weights;
3. deterministic fPINN with adaptive weights;
4. UA-fPINN deep ensemble; and
5. Bayesian fPINN when computational resources allow.

DeepXDE or an equivalent framework may facilitate implementation [15], but the fractional history operator must be checked independently against analytical functions.

7.3. Accuracy metrics

For a test set $\{z_j\}_{j=1}^{N_{\text{test}}}$, report

$$E_{L^2} = \left[\frac{\sum_j |u_{\text{pred}}(z_j) - u^\star(z_j)|^2}{\sum_j |u^\star(z_j)|^2} \right]^{1/2}, \tag{37}$$

$$E_\infty = \max_j |u_{\text{pred}}(z_j) - u^\star(z_j)|, \tag{38}$$

$$E_r = \left[\frac{1}{N_{\text{test}}} \sum_j |r_\theta(z_j)|^2 \right]^{1/2}. \tag{39}$$

For inverse parameters $\vartheta \in \{\alpha, \kappa, \rho\}$, use

$$E_\vartheta = \frac{|\widehat{\vartheta} - \vartheta^\star|}{|\vartheta^\star|}. \tag{40}$$

7.4. Uncertainty calibration

For nominal coverage q, define prediction interval coverage probability

$$\mathrm{PICP}(q) = \frac{1}{N_{\mathrm{test}}} \sum_{j=1}^{N_{\mathrm{test}}} \mathbf{1}\left\{u^\star(z_j) \in [L_q(z_j), U_q(z_j)]\right\}. \tag{41}$$

A calibrated model should satisfy $\mathrm{PICP}(q) \approx q$. Also report mean prediction interval width,

$$\mathrm{MPIW}(q) = \frac{1}{N_{\mathrm{test}}} \sum_{j=1}^{N_{\mathrm{test}}} \left[U_q(z_j) - L_q(z_j)\right], \tag{42}$$

negative log predictive density, and reliability diagrams.

7.5. Ablation studies

A complete study should isolate the contribution of each design element:

- fixed versus adaptive loss weights;
- uniform versus residual-adaptive sampling;
- exact lifting versus penalty-only boundary enforcement;
- uniform versus graded temporal grids;
- deterministic versus ensemble prediction;
- single precision versus double precision; and
- known versus inferred fractional order.

7.6. Expected qualitative behavior

Without assigning unsupported numerical values, the analysis suggests the following testable expectations. Refining the temporal grid should reduce the quadrature component until optimization or approximation error dominates. Adaptive weighting should improve simultaneous satisfaction of physics and constraints. Residual refinement should concentrate samples near $t = 0$ when fractional regularity is weak. Ensemble uncertainty should increase outside the observation region and in poorly sampled parameter regimes. These statements are hypotheses to be validated, not reported results.

8. Sensitivity and Identifiability

Inverse fractional models may be weakly identifiable because changes in α, κ, and ρ can produce similar trajectories over short observation intervals. Local sensitivity functions are

$$S_\vartheta(x,t) = \frac{\partial u(x,t;\vartheta)}{\partial \vartheta}. \tag{43}$$

The empirical Fisher information matrix is

$$F_{ij} = \sum_{n=1}^{N_d} \frac{1}{\sigma_n^2} S_{\vartheta_i}(z_n) S_{\vartheta_j}(z_n). \tag{44}$$

A large condition number of F indicates practical non-identifiability. Sensor placement should maximize a criterion such as $\det(F)$ or reduce posterior variance. Active learning based only on neural uncertainty may overlook parameter identifiability; residual and information-based criteria should therefore be combined.

Global sensitivity can be assessed using Sobol indices when uncertain parameters are independent. The total-effect index of parameter ξ_i measures its contribution, including interactions, to the variance of a quantity of interest. A trained parametric UA-fPINN can evaluate many parameter realizations cheaply after training, making it useful as a surrogate for sensitivity analysis.

9. Computational Complexity

For a time grid of size M, direct evaluation of the L1 history at all time levels costs $O(M^2)$ per spatial-parameter sample. If N_x spatial-parameter samples are used, the dominant fractional cost is $O(N_x M^2)$. Memory is $O(N_x M)$ when all network states are stored.

Potential accelerations include:

1. sum-of-exponentials approximation of the kernel;
2. FFT-based convolution when uniform grids and batching permit;
3. history compression or short-memory approximations;
4. causal temporal decomposition;
5. mixed precision for local network operations while retaining high precision in history sums; and
6. transfer learning across nearby fractional orders.

The computational comparison with classical solvers should include both training cost and amortized inference cost. A neural surrogate may be unattractive for one deterministic solve but advantageous for repeated parameter queries, inverse problems, or real-time prediction.

10. Applications in Mathematical Physics

10.1. Anomalous diffusion and porous media

Subdiffusion in heterogeneous media is a natural application because the Caputo derivative captures trapping and waiting-time effects [17]. Uncertain diffusivity fields can be represented by finite-dimensional random expansions and passed to a parametric UA-fPINN. Sparse concentration measurements can then update the uncertainty in transport parameters.

10.2. Viscoelasticity

Fractional constitutive laws interpolate between elastic and viscous behavior using fewer parameters than large integer-order rheological networks. A UA-fPINN can couple displacement equations with fractional stress–strain relations and quantify uncertainty in material orders and relaxation parameters.

10.3. Thermal processes

Fractional heat models account for memory and non-Fourier transport. Inverse estimation of thermal diffusivity and fractional order is relevant when experimental temperature data are noisy and sparse. Exact boundary lifting and calibrated uncertainty are particularly important when extrapolating beyond sensor locations.

10.4. Biological systems

Fractional reaction–diffusion and delay models can represent spatial spread with memory in epidemiology, tumor dynamics, and intracellular transport. The uncertainty-aware framework can combine mechanistic equations with incomplete clinical or laboratory measurements. Ethical and scientific interpretation requires distinguishing predictive uncertainty from biological variability.

10.5. Wave and quantum models

Space- or time-fractional wave and Schrödinger equations introduce nonlocal dispersion and anomalous propagation. In these problems, oscillatory solutions may aggravate spectral bias, so Fourier features, adaptive activations, or domain decomposition may be needed.

11. Limitations and Open Problems

The proposed framework has several limitations.

First, the theoretical estimate (27) assumes a stability inequality that must be proved for each operator and boundary condition. Nonmonotone equations, shocks, blow-up, and strongly chaotic systems may not satisfy the same bound.

Second, the L1 operator can be inaccurate for nonsmooth initial behavior. Graded meshes or corrected convolution quadrature should be considered. The temporal grid also reduces the purely mesh-free character of the method.

Third, optimization remains nonconvex. Adaptive weights and sampling improve robustness but do not guarantee convergence to a global minimum. PINN failure modes documented by [14] remain relevant.

Fourth, Bayesian neural networks can be computationally expensive and sensitive to priors. Deep ensembles are easier to implement but provide only an approximate representation of epistemic uncertainty.

Fifth, a manufactured solution validates implementation but does not replace experimental validation. Applications should compare predictions with independently measured or high-fidelity data.

Important open directions include variable-order FPDEs, space–time fractional operators, fractional equations on evolving domains, operator-learning extensions, certified posterior error bounds, multifidelity data assimilation, and active sensor placement.

12. Conclusion

This article presented an uncertainty-aware physics-informed neural network framework for nonlinear fractional partial differential equations. The method combines a neural state representation, L1-type approximation of the Caputo derivative, automatic differentiation of local spatial operators, adaptive loss balancing, residual-based collocation refinement, and probabilistic prediction. The error analysis demonstrates that a small empirical physics loss is only one component of reliability. Fractional quadrature, neural approximation, optimization, collocation sampling, and observational uncertainty must all be controlled.

A nonlinear manufactured-solution benchmark was derived for forward, parametric, and inverse experiments. The accompanying protocol specifies architecture, refinement studies, comparison methods,

accuracy metrics, calibration diagnostics, and ablation tests without inventing numerical findings. The framework is applicable to anomalous diffusion, viscoelasticity, porous media, thermal transport, biological systems, and nonlocal wave models.

The principal practical recommendation is to treat uncertainty quantification and numerical verification as integral components of fractional scientific machine learning rather than optional post-processing. Reliable UA-fPINNs should report not only predicted fields, but also residual distributions, temporal refinement behavior, sensitivity, parameter identifiability, uncertainty coverage, and comparisons with trusted discretizations.

Funding

No external funding was used in preparing this methodological study.

Conflict of Interest

The authors declare that there is no conflict of interest.

Data Availability

The data used in this study are available from the corresponding author upon reasonable request.

References

[1] Caputo, M. (1967). Linear models of dissipation whose Q is almost frequency independent–II. *Geophysical Journal International, 13*(5), 529–539.

[2] Cuomo, S., Di Cola, V. S., Giampaolo, F., Rozza, G., Raissi, M., & Piccialli, F. (2022). Scientific machine learning through physics-informed neural networks: Where we are and what's next. *Journal of Scientific Computing, 92*, 88.

[3] Diethelm, K., & Ford, N. J. (2002). Analysis of fractional differential equations. *Journal of Mathematical Analysis and Applications, 265*(2), 229–248.

[4] Diethelm, K., Ford, N. J., & Freed, A. D. (2002). A predictor-corrector approach for the numerical solution of fractional differential equations. *Nonlinear Dynamics, 29*(1–4), 3–22.

[5] Diethelm, K. (2010). *The Analysis of Fractional Differential Equations*. Springer.

[6] Gao, G.-H., Sun, Z.-Z., & Zhang, H.-W. (2014). A new fractional numerical differentiation formula to approximate the Caputo fractional derivative and its applications. *Journal of Computational Physics, 259*, 33–50.

[7] Han, J., Jentzen, A., & E, W. (2018). Solving high-dimensional partial differential equations using deep learning. *Proceedings of the National Academy of Sciences, 115*(34), 8505–8510.

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

[9] Jagtap, A. D., Kharazmi, E., & Karniadakis, G. E. (2020). Conservative physics-informed neural networks on discrete domains for conservation laws: Applications to forward and inverse problems. *Computer Methods in Applied Mechanics and Engineering, 365*, 113028.

[10] Jin, B., Lazarov, R., & Zhou, Z. (2016). An analysis of the L1 scheme for the subdiffusion equation with nonsmooth data. *IMA Journal of Numerical Analysis, 36*(1), 197–221.

[11] Juraev, D. A., & Noeiaghdam, S. (2022). Modern problems of mathematical physics and their applications. *Axioms, 11*(2), 45.

[12] Karniadakis, G. E., Kevrekidis, I. G., Lu, L., Perdikaris, P., Wang, S., & Yang, L. (2021). Physics-informed machine learning. *Nature Reviews Physics, 3*, 422–440.

[13] Kilbas, A. A., Srivastava, H. M., & Trujillo, J. J. (2006). *Theory and Applications of Fractional Differential Equations*. Elsevier.

[14] Krishnapriyan, A., Gholami, A., Zhe, S., Kirby, R., & Mahoney, M. W. (2021). Characterizing possible failure modes in physics-informed neural networks. *Advances in Neural Information Processing Systems, 34*, 26548–26560.

[15] Lu, L., Meng, X., Mao, Z., & Karniadakis, G. E. (2021). DeepXDE: A deep learning library for solving differential equations. *SIAM Review, 63*(1), 208–228.

[16] Lubich, C. (1986). Discretized fractional calculus. *SIAM Journal on Mathematical Analysis, 17*(3), 704–719.

[17] Metzler, R., & Klafter, J. (2000). The random walk's guide to anomalous diffusion: A fractional dynamics approach. *Physics Reports, 339*(1), 1–77.

[18] Mishra, S., & Molinaro, R. (2023). Estimates on the generalization error of physics-informed neural networks for approximating PDEs. *IMA Journal of Numerical Analysis, 43*(1), 1–43.

[19] Pang, G., Lu, L., & Karniadakis, G. E. (2019). fPINNs: Fractional physics-informed neural networks. *SIAM Journal on Scientific Computing, 41*(4), A2603–A2626.

[20] Podlubny, I. (1999). *Fractional Differential Equations*. Academic Press.

[21] Raissi, M., Perdikaris, P., & Karniadakis, G. E. (2019). Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics, 378*, 686–707.

[22] Sirignano, J., & Spiliopoulos, K. (2018). DGM: A deep learning algorithm for solving partial differential equations. *Journal of Computational Physics, 375*, 1339–1364.

[23] Stynes, M., O'Riordan, E., & Gracia, J. L. (2017). Error analysis of a finite difference method on graded meshes for a time-fractional diffusion equation. *SIAM Journal on Numerical Analysis, 55*(2), 1057–1079.

[24] Wang, S., Teng, Y., & Perdikaris, P. (2021). Understanding and mitigating gradient flow pathologies in physics-informed neural networks. *SIAM Journal on Scientific Computing, 43*(5), A3055–A3081.

[25] Wang, S., Yu, X., & Perdikaris, P. (2022). When and why PINNs fail to train: A neural tangent kernel perspective. *Journal of Computational Physics, 449*, 110768.

[26] Yang, Y., & Perdikaris, P. (2019). Adversarial uncertainty quantification in physics-informed neural networks. *Journal of Computational Physics, 394*, 136–152.

[27] Yang, L., Meng, X., & Karniadakis, G. E. (2021). B-PINNs: Bayesian physics-informed neural networks for forward and inverse PDE problems with noisy data. *Journal of Computational Physics, 425*, 109913.

[28] Yuste, S. B., & Acedo, L. (2005). An explicit finite difference method and a new von Neumann-type stability analysis for fractional diffusion equations. *SIAM Journal on Numerical Analysis, 42*(5), 1862–1874.

[29] Zhang, D., Lu, L., Guo, L., & Karniadakis, G. E. (2019). Quantifying total uncertainty in physics-informed neural networks for solving forward and inverse stochastic problems. *Journal of Computational Physics, 397*, 108850.

[30] Lin, Y., & Xu, C. (2007). Finite difference/spectral approximations for the time-fractional diffusion equation. *Journal of Computational Physics, 225*(2), 1533–1552.

[31] De Ryck, T., Jagtap, A. D., & Mishra, S. (2024). Error estimates for physics-informed neural networks approximating the Navier–Stokes equations. *IMA Journal of Numerical Analysis, 44*(1), 83–119.