Results in Engineering 32 (2026) 113375

Contents lists available at ScienceDirect

Results in Engineering

journal homepage: www.sciencedirect.com/journal/results-in-engineering

Research paper

A singularity-inferred operator-consistent fractional PINN for one- and two-dimensional third-kind Caputo–Hadamard auto-convolution Volterra equations with weakly singular kernels

Essaid Aourir [a,*], Hojatollah Laeli Dastjerdi [b], Kamal Shah [c], Thabet Abdeljawad [c,d,e]

[a] Department of Mathematics, Faculty of Science, Ibn Zohr University, Agadir, Morocco
[b] Department of Mathematics Education, Farhangian University, P. O. Box 14665–889, Tehran, Iran
[c] Department of Mathematics and Sciences, Prince Sultan University, Riyadh, 11586, Saudi Arabia
[d] Department of Fundamental Sciences, Faculty of Engineering and Architecture, Istanbul Gelisim University, Avcilar, 34310, Istanbul, Turkey
[e] Department of Mathematics and Applied Mathematics, School of Science and Technology, Sefako Makgatho Health Sciences University, Ga-Rankuwa, 0208, South Africa

ARTICLE INFO

Keywords:
Caputo–Hadamard derivative
Third-kind Volterra equation
Auto-convolution
Weakly singular kernel
Fractional physics-informed neural network
Gauss–Jacobi quadrature

ABSTRACT

We propose a singularity-inferred operator-consistent fractional physics-informed neural network for one- and two-dimensional third-kind Caputo–Hadamard auto-convolution Volterra equations with weakly singular kernels. The logarithmic change of variable transforms the Caputo–Hadamard derivative into a Caputo derivative, and a derivative-free Gauss–Jacobi rule is used for its numerical evaluation. The trial function contains a trainable endpoint coordinate, a polynomial term, cubic polyharmonic-spline terms, and a smooth correction. The same endpoint coordinate is used in the fitted quadrature rules. The numerical examples use an independent integral residual and higher-order evaluations of the Caputo and auto-convolution operators. In the controlled one-dimensional comparison, the proposed approximation gives the smallest mean validation error, and the two-dimensional example recovers the predicted endpoint scale.

1. Introduction

Auto-convolution Volterra equations contain the unknown function at complementary arguments and arise in nonlinear memory models. In the quadratic case the characteristic factor is $v(t-s)v(s)$. Such equations have been studied by regularization, collocation, Galerkin, and discontinuous Galerkin techniques; see, for example, [1–4]. Their analysis differs from that of standard Volterra equations because the product of two shifted solution values enters the local solvability and stability estimates [5,6].

A further difficulty is produced by third-kind degeneracy. In a third-kind equation the coefficient multiplying the differential or integral operator vanishes at an endpoint. Collocation methods for this class were analyzed in [7], while high-order recovery for nonsmooth solutions and later *hp*-collocation constructions were developed in [8,9]. Weakly singular kernels introduce an additional endpoint loss through factors of the form $(t-s)^{-\sigma}$, $0<\sigma<1$, and require quadrature or product-integration rules that incorporate the singular weight [10–12]. For Caputo–Hadamard operators, the logarithmic memory kernel produces a distinct fractional endpoint scale [13,14]. A recent piecewise shifted Jacobi method for third-kind nonlinear Caputo–Hadamard integro-differential equations is given in [15], and a fully discrete RBF collocation treatment of one- and two-dimensional third-kind Caputo–Hadamard delay integro-differential equations is developed in [16].

Fractional derivatives and Volterra integral terms are used in models with memory and hereditary effects. Weakly singular kernels also occur in these equations and require numerical treatment near the endpoint. In this paper, manufactured examples are used to study the numerical method for third-kind degeneracy and a weak singularity.

Related third-kind auto-convolution equations have been solved by moving least squares, meshless RBF collocation, and Bernstein approximation [17–19]. The equation considered here also contains a Caputo–Hadamard derivative and a weakly singular auto-convolution kernel. The numerical method uses an endpoint-adapted trial function, fitted Gauss–Jacobi rules, training and higher-order evaluations of the Caputo and auto-convolution operators, and a separate integral residual. The RBF collocation method in [16] uses the logarithmic integral form and a fully discrete error analysis. Here the solution is represented by a

* Corresponding author: essaid.aourir@edu.uiz.ac.ma
E-mail addresses: essaid.aourir@edu.uiz.ac.ma (E. Aourir), h.laeli@cfu.ac.ir (H. Laeli Dastjerdi), kamalshah408@gmail.com(kshah@psu.edu.sa) (K. Shah), tabdeljawad@psu.edu.sa (T. Abdeljawad).

https://doi.org/10.1016/j.rineng.2026.113375
Received 26 August 2026; Received in revised form 22 September 2026; Accepted 4 October 2026
Available online 6 October 2026
2590-1230/© 2026 The Author(s). Published by Elsevier B.V. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

physics-informed trial function and the weak auto-convolution kernel is included explicitly.

The one-dimensional problem class treated in this paper is the third-kind fractional auto-convolution equation with a weakly singular kernel

$$(x-1)^{\beta}\,{}^{\mathrm{CH}}_{1}D^{\mu}_{x}v(x) = a(x)v(x) + g(x) + \int_{1}^{x} \frac{\psi(x,s)}{(x-s)^{\sigma}} v(x-s+1)v(s)\,\mathrm{d}s, \quad 1 < x \le e, \tag{1.1}$$

subject to $v(1) = 0$, where

$$0 < \mu < 1, \quad 0 \le \beta < \mu, \quad 0 < \sigma < 1. \tag{1.2}$$

Here a is the linear coefficient and ψ is a regular kernel coefficient on the Volterra triangle. The factor $(x-1)^{\beta}$ gives the third-kind degeneracy, $v(x-s+1)v(s)$ is the translated quadratic auto-convolution term, and $(x-s)^{-\sigma}$ carries the explicit weak singularity.

Let $t = x-1$ and $r = s-1$, and define $\widetilde{v}(t) = v(1+t)$. Then $x-s+1 = 1+(t-r)$ and

$$v(x-s+1)v(s) = \widetilde{v}(t-r)\widetilde{v}(r).$$

The same structure is extended to the two-dimensional third-kind fractional auto-convolution equation

$$(x-1)^{\beta}{}^{\mathrm{CH}}_{1}D^{\mu}_{x}u(x,y) = a(x,y)u(x,y) + G(x,y) + \int_{1}^{x}\int_{0}^{y} \frac{\Psi(x,y,\xi,\eta)}{(x-\xi)^{\sigma}} u(x-\xi+1, y-\eta) \times u(\xi,\eta)\,\mathrm{d}\eta\,\mathrm{d}\xi, \tag{1.3}$$

on $[1,e]\times[0,1]$, with $u(1,y) = 0$. The Caputo–Hadamard singularity acts in the x-direction, whereas the auto-convolution is complementary in the translated x- and y-variables. The particular one- and two-dimensional manufactured problems used for verification are specified only in Section 6.

Physics-informed neural networks impose the governing equation through residual minimization [20,21]. Fractional PINNs require a separate numerical approximation of the nonlocal fractional operator [22]. Monte Carlo and quasi-Monte Carlo formulations have been used for high-dimensional or irregular-domain fractional problems [23,24], while deterministic weighted quadrature can substantially reduce the fractional-operator error [25]. Residual-based enrichment and residual-gradient point motion provide complementary procedures for redistributing collocation points toward difficult regions [26,27].

Auxiliary-output fPINNs have been used for fractional partial integro-differential equations [28]. Spectral-fPINNs use Jacobi polynomial representations [29], and tensor neural networks use singular trial factors together with Gauss–Jacobi quadrature [30]. Physics-informed radial-basis networks provide another basis-function form of the approximation [31,32]. Approximation results for RBF networks are given in [33], while polyharmonic-spline approximation is considered separately below. PINNIES uses numerical quadrature for Volterra, Fredholm, and fractional operators [34]. Kernel-based PINNs and transformed fractional PINNs provide other numerical treatments of nonlocal operators [35,36]. Fractional PINNs with singular terms or singular features are studied in [37,38].

Recent neural methods for fractional differential equations include Hermite interpolation for time-dependent Caputo equations [39], Kolmogorov–Arnold networks for time-fractional equations [40], and wavelet neural networks for fractional differential equations [41]. These references concern fractional differential equations; the third-kind Caputo–Hadamard auto-convolution equation is not considered in those papers.

The method used here combines RBF-type approximation, a singular endpoint coordinate, Gauss–Jacobi quadrature, adaptive points, and a fractional PINN formulation. We set $\rho = \log x$ and use $z = \rho^{\alpha_\Theta}$ in the trial function. The Caputo term is evaluated by the derivative-free identity, and the current endpoint parameter is used in the fitted Jacobi weights. A separate quadrature is used for the equivalent integral residual, while higher quadrature orders are used for the reference Caputo and auto-convolution evaluations. We denote this method by SI-OC-PHS-fPINN.

Moving least squares, meshless RBF collocation, and Bernstein approximation have been applied to third-kind auto-convolution Volterra equations [17–19]. Piecewise shifted Jacobi polynomials and fully discrete RBF collocation have been used for third-kind Caputo–Hadamard equations [15,16]. In fPINNs, the fractional operator is evaluated numerically [22]. Spectral-fPINNs use Jacobi polynomials [29], and the tensor-neural-network method in [30] uses a fractional-power trial function and Gauss–Jacobi quadrature. The method considered in this paper uses a trainable endpoint coordinate, the Caputo–Hadamard operator, a weakly singular auto-convolution term, fitted Gauss–Jacobi rules, and the equivalent Volterra equation in both one and two dimensions.

The equivalent Volterra equation is used in the analysis. We prove local existence and uniqueness for an algebraic source and determine the corresponding endpoint power. The contraction estimate and the Gauss–Jacobi error estimate give residual stability and an error bound containing approximation, quadrature, and optimization terms. These results apply on a local interval where the contraction condition holds.

The main contributions are summarized as follows.

1. A derivative-free Caputo representation in logarithmic coordinates is combined with singularity-fitted Gauss–Jacobi product integration for third-kind Caputo–Hadamard auto-convolution equations with weakly singular kernels.
2. In this paper, we aim to develop an efficient method for accurately approximating integral operators, as well as fractional derivative operators, for solving mathematical problems involving these operators.
3. Then Gauss-Jacobi quadrature is used to discretize Caputo fractional derivatives.
4. Some numerical experiments are given to illustrate the theoretical results.

This paper is organized as follows. Section 2: presents the existence and uniqueness of solutions. Subsequently, in Section 3, we introduce the method for approximating various integral and fractional operators. Section 4: describes the proposed scheme. Section 5: proposes a convergence analysis for the proposed approach. Section 6 reports the numerical experiments, followed by the discussion and conclusions.

2. Mathematical formulation and local analysis

This section gives the fractional operators, the logarithmic Volterra form, and the local existence and endpoint results used in the stability analysis.

2.1. Fractional preliminaries

The Hadamard fractional integral and Caputo–Hadamard derivative are standard; see [13–15].

Definition 2.1 (Hadamard fractional integral). Let $\mu > 0$ and let f be measurable on $(1,T]$. At every $x \in (1,T]$ for which

$$\int_{1}^{x} \left(\log\frac{x}{s}\right)^{\mu-1} |f(s)|\,\frac{\mathrm{d}s}{s} < \infty,$$

the left Hadamard fractional integral of order μ is defined by

$$ {}^{\mathrm{H}}_{1}I^{\mu}_{x} f(x) = \frac{1}{\Gamma(\mu)} \int_{1}^{x} \left(\log\frac{x}{s}\right)^{\mu-1} f(s)\,\frac{\mathrm{d}s}{s}. \tag{2.1}$$

For continuous f the integral is finite for every $x \in [1,T]$, with value zero at $x = 1$.

For $\vartheta > -1$, the logarithmic power rule is

$$ {}^{\mathrm{H}}_{1}I^{\mu}_{x}(\log x)^{\vartheta} = \frac{\Gamma(\vartheta+1)}{\Gamma(\vartheta+\mu+1)}(\log x)^{\vartheta+\mu}. \tag{2.2}$$

Definition 2.2 (Caputo–Hadamard derivative). Let $0 < \mu < 1$, put $L = \log T$, and set $w(\rho) = v(e^\rho)$. Assume $w \in AC([0, L])$. For $\rho \in (0, L]$ such that

$$(\rho - \tau)^{-\mu} w'(\tau) \in L^1(0, \rho),$$

the left Caputo–Hadamard derivative at $x = e^\rho$ is

$$ {}^{\mathrm{CH}}_{1}D^{\mu}_{x} v(x) = \frac{1}{\Gamma(1-\mu)} \int_1^x \Big(\log \frac{x}{s}\Big)^{-\mu} \Big(s \frac{\mathrm{d}}{\mathrm{d}s} v(s)\Big) \frac{\mathrm{d}s}{s}. \tag{2.3}$$

Equivalently, this is $D^{\mu}_{C} w(\rho)$. The definition is understood almost everywhere for an absolutely continuous w, and pointwise wherever the displayed weighted integral is finite.

Under these hypotheses, the standard Caputo composition identity gives, for the continuous representative,

$$ {}^{\mathrm{H}}_{1}I^{\mu}_{x}\, {}^{\mathrm{CH}}_{1}D^{\mu}_{x} v(x) = v(x) - v(1). \tag{2.4}$$

For $\vartheta > 0$,

$$ {}^{\mathrm{CH}}_{1}D^{\mu}_{x} (\log x)^{\vartheta} = \frac{\Gamma(\vartheta+1)}{\Gamma(\vartheta-\mu+1)} (\log x)^{\vartheta-\mu}. \tag{2.5}$$

The logarithmic change of variables

$$\rho = \log x, \quad w(\rho) = v(e^\rho) \tag{2.6}$$

maps the Caputo–Hadamard derivative to the ordinary Caputo derivative:

$$ {}^{\mathrm{CH}}_{1}D^{\mu}_{x} v(x) = D^{\mu}_{C} w(\rho). \tag{2.7}$$

This identity is the starting point of the numerical construction.

The beta function is written as

$$\mathrm{B}(p, q) = \int_0^1 t^{p-1}(1-t)^{q-1}\, \mathrm{d}t = \frac{\Gamma(p)\Gamma(q)}{\Gamma(p+q)}, \quad p, q > 0. \tag{2.8}$$

It is used repeatedly in the local solvability and endpoint estimates.

For Eq. (1.1), set $x = e^\rho$ and $s = e^\eta$. This change of variable transforms the Caputo–Hadamard term to the Caputo form and gives a Volterra equation in ρ. For the one-dimensional Eq. (1.1), put $x = e^\rho$ and $s = e^\eta$. The auto-convolution operator becomes

$$J[w](\rho) = \int_0^\rho \frac{\psi(e^\rho, e^\eta) e^\eta}{(e^\rho - e^\eta)^{\sigma}} u(\log(1 + e^\rho - e^\eta)) w(\eta)\, \mathrm{d}\eta. \tag{2.9}$$

In the transformed equations below, the notation is shortened by writing $a(\rho) := a(e^\rho)$ and $g(\rho) := g(e^\rho)$; the same convention is used for manufactured source functions. This convention changes only the argument, not the coefficient or source values. Using (2.7), the transformed equation is

$$(e^\rho - 1)^{\beta} D^{\mu}_{C} w(\rho) = a(\rho) w(\rho) + g(\rho) + J[w](\rho), \quad w(0) = 0. \tag{2.10}$$

Applying the Riemann–Liouville integral of order μ to (2.10) yields

$$(\mathcal{T} w)(\rho) = \frac{1}{\Gamma(\mu)} \int_0^\rho (\rho - s)^{\mu-1} (e^s - 1)^{-\beta} \big[a(s) w(s) + g(s) + J[w](s)\big]\, \mathrm{d}s. \tag{2.11}$$

A fixed point $w = \mathcal{T} w$, $w(0) = 0$, is a mild solution. Whenever the right-hand side has the regularity required by the Caputo derivative, (2.11) and (2.10) are equivalent.

2.2. Existence and uniqueness

For $h \in (0, 1]$, let

$$\Delta_h := \{(x, s) : 1 \le s \le x \le e^h\},$$

and define

$$A_h := \max_{1 \le x \le e^h} |a(x)|, \quad \psi^*_h := \max_{(x,s) \in \Delta_h} |\psi(x, s)|, \quad M_h := e^h \psi^*_h. \tag{2.12}$$

Since $e^\rho - e^\eta \ge \rho - \eta$ for $0 \le \eta \le \rho \le h$, $e^\eta \le e^h$, and the shifted argument in (2.9) belongs to $[0, \rho]$, any $w \in C[0, h]$ with $\|w\|_\infty \le R$ satisfies

$$|J[w](\rho)| \le \frac{M_h R^2}{1-\sigma} \rho^{1-\sigma}. \tag{2.13}$$

Moreover, if $\|w\|_\infty, \|z\|_\infty \le R$,

$$|J[w](\rho) - J[z](\rho)| \le \frac{2 M_h R}{1-\sigma} \rho^{1-\sigma} \|w - z\|_\infty. \tag{2.14}$$

Theorem 2.3. *Assume (1.2), $a \in C([1, e^h])$, and $\psi \in C(\Delta_h)$. Suppose that, for some $\gamma > \beta - \mu$,*

$$g(\rho) = \rho^\gamma \widehat{g}(\rho), \quad 0 < \rho \le h, \tag{2.15}$$

where $\widehat{g} \in C[0, h]$, and set $G = \|\widehat{g}\|_\infty$. Let $B_R = \{w \in C[0, h] : w(0) = 0,\ \|w\|_\infty \le R\}$. If

$$\frac{1}{\Gamma(\mu)} \Big[A_h R \mathrm{B}(1-\beta, \mu) h^{\mu-\beta} + G \mathrm{B}(\gamma - \beta + 1, \mu) h^{\mu+\gamma-\beta} + \frac{M_h R^2}{1-\sigma} \mathrm{B}(2 - \sigma - \beta, \mu) h^{\mu+1-\sigma-\beta} \Big] \le R, \tag{2.16}$$

and

$$\kappa_h := \frac{1}{\Gamma(\mu)} \Big[A_h \mathrm{B}(1-\beta, \mu) h^{\mu-\beta} + \frac{2 M_h R}{1-\sigma} \mathrm{B}(2 - \sigma - \beta, \mu) h^{\mu+1-\sigma-\beta} \Big] < 1, \tag{2.17}$$

then (2.11) has a unique fixed point in B_R.

Proof. Because $e^s - 1 \ge s$, one has $(e^s - 1)^{-\beta} \le s^{-\beta}$. Using (2.13) in (2.11) and the identity

$$\int_0^\rho (\rho - s)^{\mu-1} s^q\, \mathrm{d}s = \rho^{\mu+q} \mathrm{B}(q+1, \mu), \quad q > -1,$$

gives (2.16). The conditions $\beta < \mu$ and $\gamma > \beta - \mu$ imply that all displayed powers of ρ are positive. Hence $\mathcal{T}(B_R) \subset B_R$. For $w, z \in B_R$, (2.14) gives

$$\|\mathcal{T} w - \mathcal{T} z\|_\infty \le \kappa_h \|w - z\|_\infty.$$

□

The leading behavior at $\rho = 0$ follows from (2.11), the fractional power rule, and the beta function [13,14]. The third-kind factor contributes the power $-\beta$, while the right-hand side contributes the power γ.

Proposition 2.5 (Endpoint law). *Let w be the local solution from Theorem 2.3. Assume that*

$$a(\rho) w(\rho) + g(\rho) + J[w](\rho) = b_0 \rho^\gamma + o(\rho^\gamma), \quad \gamma > \beta - \mu. \tag{2.18}$$

Then

$$w(\rho) = b_0 \frac{\Gamma(\gamma - \beta + 1)}{\Gamma(\mu + \gamma - \beta + 1)} \rho^{\mu+\gamma-\beta} + o\big(\rho^{\mu+\gamma-\beta}\big). \tag{2.19}$$

Proof. Write the remainder in (2.18) as $s^\gamma r(s)$, where $r(s) \to 0$ as $s \downarrow 0$. In (2.11), use $(e^s - 1)^{-\beta} = s^{-\beta}(1 + O(s))$ and set $s = \rho t$. After division by $\rho^{\mu+\gamma-\beta}$, the leading term is

$$\frac{b_0}{\Gamma(\mu)} \int_0^1 (1-t)^{\mu-1} t^{\gamma-\beta}\, \mathrm{d}t.$$

Because $\gamma > \beta - \mu$, the weight $(1-t)^{\mu-1} t^{\gamma-\beta}$ is integrable on $(0, 1)$. Moreover, r is uniformly bounded on $(0, \rho]$ for sufficiently small ρ. Hence dominated convergence applies to the scaled remainder, and the beta-function identity yields (2.19). □

Proposition 2.5 gives the relation

$$\nu = \mu + \gamma - \beta, \quad \text{equivalently} \quad \gamma = \nu + \beta - \mu. \tag{2.20}$$

For the principal benchmark, $\nu = 0.5$ corresponds to $\gamma = 0.15$. For the two-dimensional benchmark, the leading forcing exponent is $\gamma = 0$, and (2.19) gives $\nu = 0.35$.

Corollary 2.6 (Two-dimensional local solvability and endpoint law). *Let $0 < Y \le 1$ and set*

$$U(\rho, y) := u(e^{\rho}, y), \quad \widetilde{a}(\rho, y) := a(e^{\rho}, y), \quad \widetilde{G}(\rho, y) := G(e^{\rho}, y).$$

Assume that, for some $\gamma > \beta - \mu$,

$$\widetilde{G}(\rho, y) = \rho^{\gamma} \widehat{G}(\rho, y), \quad \widehat{G} \in C([0, h] \times [0, Y]),$$

and write $G_{2,h} := \|\widehat{G}\|_{\infty}$. Suppose also that a and the kernel coefficient Ψ in (1.3) are continuous on the corresponding compact domains. Define

$$\Delta_{2,h,Y} := \{(x, y, \xi, \eta) : 1 \le \xi \le x \le e^{h},\ 0 \le \eta \le y \le Y\},$$

$$A_{2,h} := \max_{1 \le x \le e^{h}, 0 \le y \le Y} |a(x, y)|,$$

$$\Psi^{*}_{2,h} := \max_{\Delta_{2,h,Y}} |\Psi(x, y, \xi, \eta)|,$$

$$M_{2,h} := Y e^{h} \Psi^{*}_{2,h}.$$

For $U \in C([0, h] \times [0, Y])$, the transformed double auto-convolution is

$$J_2[U](\rho, y) = \int_0^{\rho} \int_0^{y} \frac{\Psi(e^{\rho}, y, e^{\eta}, \zeta) e^{\eta}}{(e^{\rho} - e^{\eta})^{\sigma}} U(\log(1 + e^{\rho} - e^{\eta}), y - \zeta) \times U(\eta, \zeta)\, \mathrm{d}\zeta\, \mathrm{d}\eta.$$

If $\|U\|_{\infty} \le R$, then, uniformly for $0 \le y \le Y$,

$$|J_2[U](\rho, y)| \le \frac{M_{2,h} R^2}{1 - \sigma} \rho^{1-\sigma}, \quad 0 \le \rho \le h, \tag{2.21}$$

and, for $\|U\|_{\infty}, \|V\|_{\infty} \le R$,

$$|J_2[U](\rho, y) - J_2[V](\rho, y)| \le \frac{2 M_{2,h} R}{1 - \sigma} \rho^{1-\sigma} \|U - V\|_{\infty}, \quad 0 \le \rho \le h. \tag{2.22}$$

The corresponding integral operator is

$$(\mathcal{T}_2 U)(\rho, y) = \frac{1}{\Gamma(\mu)} \int_0^{\rho} (\rho - s)^{\mu - 1} (e^{s} - 1)^{-\beta} \times \left[\widetilde{a}(s, y) U(s, y) + \widetilde{G}(s, y) + J_2[U](s, y)\right] \mathrm{d}s. \tag{2.23}$$

Let

$$\mathcal{B}^{(2)}_{R} := \{U \in C([0, h] \times [0, Y]) : U(0, y) = 0,\ \|U\|_{\infty} \le R\}.$$

If

$$\frac{1}{\Gamma(\mu)} \left[A_{2,h} R\, \mathrm{B}(1 - \beta, \mu) h^{\mu - \beta} + G_{2,h}\, \mathrm{B}(\gamma - \beta + 1, \mu) h^{\mu + \gamma - \beta} + \frac{M_{2,h} R^2}{1 - \sigma} \mathrm{B}(2 - \sigma - \beta, \mu) h^{\mu + 1 - \sigma - \beta}\right] \le R, \tag{2.24}$$

and

$$\kappa_{2,h} := \frac{1}{\Gamma(\mu)} \left[A_{2,h} \mathrm{B}(1 - \beta, \mu) h^{\mu - \beta} + \frac{2 M_{2,h} R}{1 - \sigma} \mathrm{B}(2 - \sigma - \beta, \mu) h^{\mu + 1 - \sigma - \beta}\right] < 1, \tag{2.25}$$

then the two-dimensional integral formulation has a unique fixed point in $\mathcal{B}^{(2)}_{R}$. Moreover, if $b_0 \in C([0, Y])$ and, uniformly in y,

$$\widetilde{a}(\rho, y) U(\rho, y) + \widetilde{G}(\rho, y) + J_2[U](\rho, y) = b_0(y) \rho^{\gamma} + o(\rho^{\gamma}), \quad \rho \downarrow 0, \tag{2.26}$$

then

$$U(\rho, y) = b_0(y) \frac{\Gamma(\gamma - \beta + 1)}{\Gamma(\mu + \gamma - \beta + 1)} \rho^{\mu + \gamma - \beta} + o\big(\rho^{\mu + \gamma - \beta}\big), \tag{2.27}$$

uniformly for $y \in [0, Y]$.

Proof. Because $e^{\rho} - e^{\eta} \ge \rho - \eta$ and $e^{\eta} \le e^{h}$ for $0 \le \eta \le \rho \le h$, the y-integration contributes at most the factor Y, and (2.21) follows from $\int_0^{\rho} (\rho - \eta)^{-\sigma}\, \mathrm{d}\eta = \rho^{1-\sigma}/(1 - \sigma)$. The product difference $|U_1 U_2 - V_1 V_2| \le R|U_1 - V_1| + R|U_2 - V_2|$ gives the corresponding Lipschitz estimate. Applying the same beta-integral bounds as in Theorem 2.3, uniformly in y, yields (2.24) and the contraction constant (2.25); Banach's fixed-point theorem gives existence and uniqueness. Finally, write the uniform remainder in (2.26) as $\rho^{\gamma} r(\rho, y)$ with $\sup_{0 \le y \le Y} |r(\rho, y)| \to 0$. After setting $s = \rho t$, the scaled integrand is dominated, uniformly in y, by a constant multiple of $(1 - t)^{\mu - 1} t^{\gamma - \beta}$, which is integrable because $\gamma > \beta - \mu$. Dominated convergence therefore applies uniformly in y, and the beta-function identity gives (2.27). □

3. Operator-consistent numerical formulation

The transformed equation contains the Caputo kernel and the weakly singular auto-convolution kernel. The Caputo term is treated by a weighted Gauss–Jacobi rule, the auto-convolution term by product integration, and the equivalent integral equation supplies the residual used in the stability estimate.

First, the Caputo derivative is evaluated through function differences rather than through a shifted automatic derivative of the approximation. The identity below is obtained from the standard Caputo integral representation [13,14] by an integration-by-parts argument.

Proposition 3.1 (Derivative-free Caputo identity). *Let $0 < \mu < 1$, $\rho > 0$, and assume $w \in AC([0, \rho])$ and $w \in C^1([\rho - \delta, \rho])$ for some $\delta \in (0, \rho]$. Then*

$$D^{\mu}_{C} w(\rho) = \frac{\rho^{-\mu}}{\Gamma(1 - \mu)} \left[w(\rho) - w(0) + \mu \int_0^1 q^{-\mu} \frac{w(\rho) - w(\rho(1 - q))}{q}\, \mathrm{d}q\right]. \tag{3.1}$$

Proof. Starting from the Caputo definition, put $r = \rho - \tau$ and $h(r) = w(\rho) - w(\rho - r)$. The function h is absolutely continuous and $h'(r) = w'(\rho - r)$ almost everywhere. Integration by parts on $[\varepsilon, \rho]$ gives

$$\int_{\varepsilon}^{\rho} r^{-\mu} h'(r)\, \mathrm{d}r = \rho^{-\mu}[w(\rho) - w(0)] - \varepsilon^{-\mu} h(\varepsilon) + \mu \int_{\varepsilon}^{\rho} r^{-\mu - 1} h(r)\, \mathrm{d}r.$$

Because w is C^1 near ρ, one has $h(r) = r w'(\rho) + o(r)$ as $r \downarrow 0$, and therefore $\varepsilon^{-\mu} h(\varepsilon) \to 0$. Letting $\varepsilon \downarrow 0$ and using $r = \rho q$ yields (3.1). □

For $\rho > 0$, the assumptions allow w' to be singular at the left endpoint and require C^1 regularity only near the evaluation point. For the trial functions, the quotient in (3.1) has the limit $\rho w'(\rho)$ at $q = 0$. Gauss–Jacobi quadrature is applied directly to the weight $q^{-\mu}$, so only values of the trial function are required at the shifted quadrature points [36].

For the one-dimensional weakly singular auto-convolution term, set $s = 1 + (x - 1)t$. After the factor $(1 - t)^{-\sigma}$ is isolated, write the remaining factor as $C(t)$. For any a_q satisfying $a_q > -1$ and $a_q - \sigma > -1$,

$$\int_0^1 (1 - t)^{-\sigma} C(t)\, \mathrm{d}t = \int_0^1 t^{a_q} (1 - t)^{a_q - \sigma} \frac{C(t)}{t^{a_q} (1 - t)^{a_q}}\, \mathrm{d}t. \tag{3.2}$$

Eq. (3.2) is an exact product-integration form of the weakly singular integral. Product integration for nonlinear weakly singular Volterra equations is studied in [12]. The fitted exponent is taken from the endpoint factor in the trial function. Write $w_{\Theta}(r) = r^{\alpha_{\Theta}} F_{\Theta}(r^{\alpha_{\Theta}})$. For fixed $\rho > 0$ and $d = e^{\rho} - 1$, let

$$r_0(t) = \log(1 + dt), \quad r_1(t) = \log(1 + d(1 - t)).$$

The ratios $r_0(t)/t$ and $r_1(t)/(1 - t)$ extend continuously to positive values at the endpoints. Therefore

$$\frac{w_{\Theta}(r_0(t)) w_{\Theta}(r_1(t))}{t^{\alpha_{\Theta}} (1 - t)^{\alpha_{\Theta}}} = \left(\frac{r_0(t)}{t}\right)^{\alpha_{\Theta}} \left(\frac{r_1(t)}{1 - t}\right)^{\alpha_{\Theta}} F_{\Theta}(r_0(t)^{\alpha_{\Theta}}) F_{\Theta}(r_1(t)^{\alpha_{\Theta}})$$

is continuous on $[0, 1]$. The endpoint powers are included in the Gauss–Jacobi weight. Higher-order estimates require the corresponding smoothness of the remaining factor, as assumed in Lemma 5.1. The same condition is used for the regular factor in the independent integral residual. In the numerical rule,

$$a_q = \mathrm{detach}(\alpha_{\Theta}) \tag{3.3}$$

is fixed during a quadrature refresh. Here detach means that the current scalar value of α_{Θ} is used to construct the Jacobi nodes and weights,

which are then held fixed when derivatives of the loss are computed until the next refresh. The current endpoint coordinate determines the quadrature weight, while gradients are not taken through the quadrature construction.

In each optimization stage, the Jacobi nodes and weights are fixed. At a refresh they are computed from the current endpoint coordinate before the next stage. The procedure uses optimization with fixed quadrature followed by an update of the quadrature rule. Section 5 applies to each fixed-stage discrete operator. In the one-dimensional example, one refresh is made between the two AdamW stages; the rules are also computed for the L-BFGS candidates, and the endpoint coordinate and centers are fixed in the final coefficient step.

Eq. (2.11) also provides an equivalent integral residual. On $[0,1]$ it can be written as

$$w(\rho) = \frac{\rho^{\mu}}{\Gamma(\mu)} \int_0^1 (1-t)^{\mu-1} H_\rho(t)\,\mathrm{d}t, \qquad H_\rho(t) := \frac{a(\rho t)w(\rho t) + g(\rho t) + J[w](\rho t)}{(e^{\rho t}-1)^{\beta}}. \tag{3.4}$$

For any $b_q > -1$ the exact identity

$$\int_0^1 (1-t)^{\mu-1} H_\rho(t)\,\mathrm{d}t = \int_0^1 t^{b_q}(1-t)^{\mu-1} \frac{H_\rho(t)}{t^{b_q}}\,\mathrm{d}t \tag{3.5}$$

permits an independently fitted left-endpoint Jacobi weight. At a quadrature refresh we use

$$b_q = \mathrm{detach}(\alpha_\Theta) - \mu. \tag{3.6}$$

Since $\alpha_\Theta > 0$ and $0 < \mu < 1$, one has $b_q > -1$. The fitted rule requires the regularity of $H_\rho(t)/t^{b_q}$. Eqs. (2.10) and (3.4) are equivalent before discretization. In the computation they use separate quadrature rules; (3.4) provides a second training residual and is used in the stability estimate of Section 5.

4. SI-OC-PHS-fPINN architecture and training algorithm

This section describes the endpoint-adapted trial function, the discrete operators, the loss function, adaptive collocation, and the optimization procedure. Fig. 1 shows the main steps. The exact solution is used to construct the manufactured source and for independent verification; it is not used in training, adaptive point selection, quadrature fitting, or model selection.

4.1. Endpoint-adapted trial space and discrete operators

The endpoint-adapted trial function uses the coordinate

$$z = \rho^{\alpha_\Theta}, \qquad \alpha_\Theta = \alpha_{\min} + (\alpha_{\max} - \alpha_{\min})\,\mathrm{sigmoid}(\alpha_{\mathrm{raw}}), \quad (\alpha_{\min}, \alpha_{\max}) = (0.15, 0.90). \tag{4.1}$$

The initial value is $\alpha_\Theta = 0.43$. The homogeneous condition is imposed exactly through

$$w_\Theta(\rho) = z F_\Theta(z). \tag{4.2}$$

The regular factor is

$$F_\Theta(z) = P_6(z) + s_{\mathrm{phs}} \frac{1}{\sqrt{m}} \sum_{j=1}^{m} \omega_j |z - c_j|^3 + s_{\mathrm{corr}} N_\Theta(z), \quad m = 34. \tag{4.3}$$

Here P_6 is a degree-six polynomial, $\phi(r) = r^3$ is the one-dimensional cubic polyharmonic spline, and N_Θ is a two-hidden-layer tanh network with 18 units in each layer. The cubic PHS is scale-free and has no shape parameter. Approximation results for RBF networks are given in [33], and physics-informed radial-basis networks are considered in [31,32]. Since the basis used here is a cubic PHS, the relevant approximation background is provided by polyharmonic-spline theory [42–45].

The PHS centers are trainable. To preserve their order, positive spacings are generated by a softmax map and accumulated:

$$d_j = \frac{\exp(\zeta_j)}{\sum_{k=1}^{m+1} \exp(\zeta_k)}, \quad c_j = \varepsilon + (1-2\varepsilon)\sum_{k=1}^{j} d_k, \quad j = 1, \dots, m, \tag{4.4}$$

with a small fixed $\varepsilon > 0$. The trainable parameters are the endpoint exponent, the ordered center locations, the PHS weights, the polynomial coefficients, the correction-network parameters, and two amplitudes. The cubic PHS provides a scale-free basis in the endpoint coordinate.

The parameter α_Θ is an internal coordinate parameter and is not identified a priori with the physical endpoint exponent. If, for some integer $\ell \geq 0$,

$$F_\Theta(z) = c_\ell z^{\ell} + o(z^{\ell}), \quad c_\ell \neq 0,$$

then, by (4.2),

$$w_\Theta(\rho) \sim c_\ell \rho^{(\ell+1)\alpha_\Theta}. \tag{4.5}$$

The endpoint regularity is measured independently by

$$\widehat{\nu}_{\mathrm{eff}} = \mathrm{slope}\big(\log|w_\Theta(\rho)|, \log\rho\big) \tag{4.6}$$

on a fixed near-endpoint interval.

Proposition 3.1 is discretized by Gauss–Jacobi quadrature. Let $\{q_j, \omega_j^C\}_{j=1}^{Q_C}$ be the nodes and weights on $[0,1]$ for the weight $q^{-\mu}$. Then

$$D^{\mu}_{C,Q_C} w_\Theta(\rho) = \frac{\rho^{-\mu}}{\Gamma(1-\mu)} \left[w_\Theta(\rho) - w_\Theta(0) + \mu \sum_{j=1}^{Q_C} \omega_j^C \frac{w_\Theta(\rho) - w_\Theta(\rho(1-q_j))}{q_j} \right]. \tag{4.7}$$

Only trial-function values are required at shifted quadrature points.

For the auto-convolution term in (1.1), write $d = x - 1$ and set $s = 1 + dt$. Then

$$J[w_\Theta](\rho) = d^{1-\sigma} \int_0^1 (1-t)^{-\sigma} C_\rho(t)\,\mathrm{d}t, \tag{4.8}$$

where

$$C_\rho(t) = \psi(1+d, 1+dt)\, w_\Theta(\log[1+d(1-t)]) w_\Theta(\log[1+dt]). \tag{4.9}$$

At a quadrature refresh, set $a_q = \mathrm{detach}(\alpha_\Theta)$. If $\{t_j, \omega_j^V\}_{j=1}^{Q_V}$ is the Gauss–Jacobi rule for the weight $t^{a_q}(1-t)^{a_q-\sigma}$, the fitted product rule is

$$J_{Q_V}[w_\Theta](\rho) = d^{1-\sigma} \sum_{j=1}^{Q_V} \omega_j^V \frac{C_\rho(t_j)}{t_j^{a_q}(1-t_j)^{a_q}}. \tag{4.10}$$

At each stage, the quadrature parameters are obtained from the current endpoint coordinate and the nodes and weights remain fixed. The differential residual is evaluated with the training orders (Q_C, Q_V) and with the higher reference orders $(Q_C^{\mathrm{ref}}, Q_V^{\mathrm{ref}})$. The integral residual uses a separate fitted Gauss–Jacobi rule of order Q_I.

4.2. Physics loss, adaptive training, and model selection

The differential residual is

$$R_{\mathrm{dif}}(\rho) = (e^{\rho}-1)^{\beta} D^{\mu}_{C,Q_C} w_\Theta(\rho) - a(\rho) w_\Theta(\rho) - g(\rho) - J_{Q_V}[w_\Theta](\rho). \tag{4.11}$$

It is normalized by

$$R_n = \frac{R_{\mathrm{dif}}}{1 + |(e^{\rho}-1)^{\beta} D^{\mu}_{C,Q_C} w_\Theta| + |a(\rho) w_\Theta| + |g| + |J_{Q_V}[w_\Theta]|}. \tag{4.12}$$

The equivalent integral Eq. (3.4) supplies a second normalized residual,

$$R_{\mathrm{int}} = \frac{w_\Theta - \mathcal{T}^{\mathrm{int}}_{Q_I} w_\Theta}{1 + |w_\Theta| + |\mathcal{T}^{\mathrm{int}}_{Q_I} w_\Theta|}, \tag{4.13}$$

where Q_I is the independent integral-residual quadrature order and $\mathcal{T}^{\mathrm{int}}_{Q_I}$ denotes the discrete right-hand side of (3.4). Likewise, R_{ref} denotes the

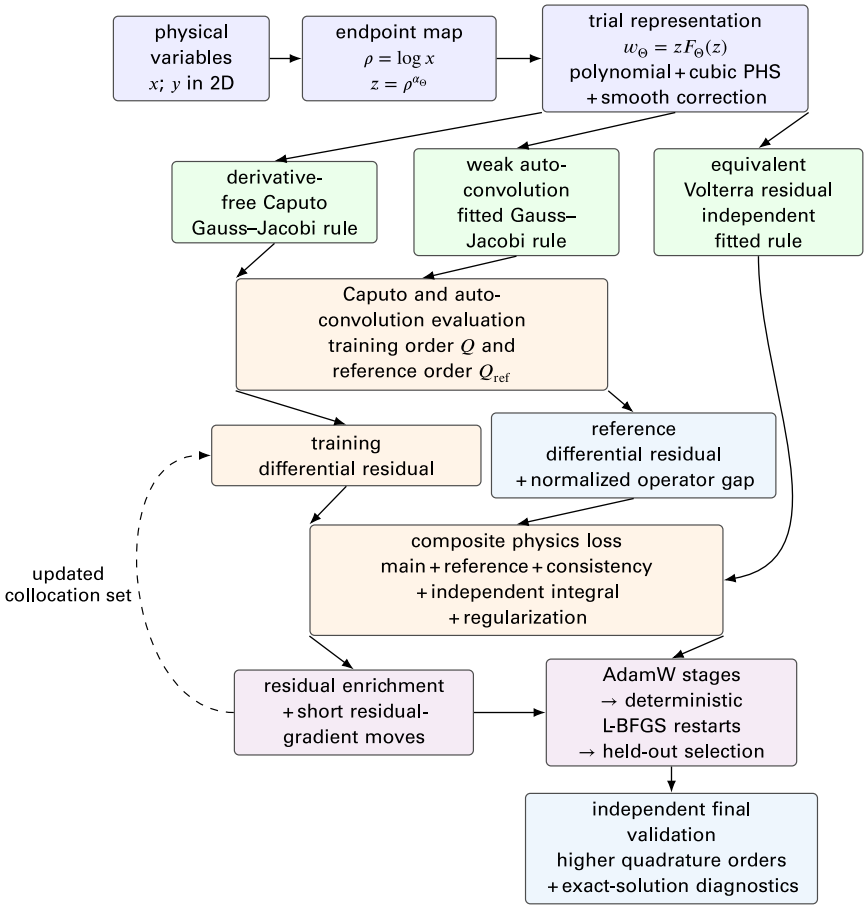


Fig. 1. Computational structure of the SI-OC-PHS-fPINN. The Caputo and auto-convolution branches are evaluated at training and higher-order reference levels. The equivalent Volterra residual is evaluated independently with its own fitted quadrature before entering the composite loss. Exact solutions are reserved for source construction and final verification.

normalized differential residual obtained from the higher-order reference Caputo and Volterra evaluations.

For a minibatch $\{\rho_i\}_{i=1}^{N_b}$ with positive adaptive weights ω_i, let $\mathcal{H}$ contain the indices of the largest 8% of the values $R_n(\rho_i)^2$. The principal residual loss implemented in the numerical code is

$$\mathcal{L}_{\mathrm{main}} = \frac{\sum_{i=1}^{N_b} \omega_i R_n(\rho_i)^2}{\sum_{i=1}^{N_b} \omega_i} + 0.45\,\frac{1}{|\mathcal{H}|}\sum_{i\in\mathcal{H}} R_n(\rho_i)^2 + 0.04\,\frac{1}{N_b}\sum_{i=1}^{N_b}\left(\frac{R_{\mathrm{dif}}(\rho_i)}{1+|g(\rho_i)|}\right)^2. \tag{4.14}$$

If D_Q, J_Q and $D_{\mathrm{ref}}, J_{\mathrm{ref}}$ denote the training and reference Caputo and Volterra evaluations, respectively, the operator-consistency term is

$$\mathcal{L}_{\mathrm{cons}} = \mathrm{mean}\left[\left(\frac{D_Q - D_{\mathrm{ref}}}{1+|D_{\mathrm{ref}}|}\right)^2\right] + \mathrm{mean}\left[\left(\frac{J_Q - J_{\mathrm{ref}}}{1+|J_{\mathrm{ref}}|}\right)^2\right]. \tag{4.15}$$

During the AdamW stages, the composite loss is

$$\mathcal{L} = \mathcal{L}_{\mathrm{main}} + 0.28\,\mathcal{L}_{\mathrm{ref}} + 0.12\,\mathcal{L}_{\mathrm{cons}} + \lambda_{\mathrm{int}}(e)\,\mathcal{L}_{\mathrm{int}} + \mathcal{L}_{\mathrm{reg}}, \tag{4.16}$$

where $\mathcal{L}_{\mathrm{ref}} = \mathrm{mean}(R_{\mathrm{ref}}^2)$, $\mathcal{L}_{\mathrm{int}} = \mathrm{mean}(R_{\mathrm{int}}^2)$, and, for an AdamW stage of N_e iterations,

$$\lambda_{\mathrm{int}}(e) = 0.18\,\min\left\{1, \frac{e + 0.15N_e}{0.45N_e + 10^{-12}}\right\}.$$

In the long L-BFGS restarts and the final coefficient polish, the reference, consistency, and integral coefficients are 0.8(0.28), 0.8(0.12), and 0.70(0.18), respectively. The mathematical operators and trial space are unchanged across these optimization stages. The regularization term controls center spacing and the auxiliary smooth branch. Exact-solution values do not enter the training loss.

Training starts with low-discrepancy points and points graded toward $\rho = 0$. After the first optimization stage, $|R_n|$ is evaluated on a larger candidate set. Half of the new points are taken from the largest residuals and the remaining points are sampled from a residual-weighted distribution with a uniform part [26]. A small group of points is moved by two short residual-gradient steps [27].

The reported one-dimensional calculation uses 1600 AdamW steps, followed by a quadrature refresh and residual-based enrichment, and then 850 additional AdamW steps. Three deterministic L-BFGS restarts are started from the same selected Adam checkpoint. Their candidates are compared on held-out physics through

$$S_{\mathrm{hold}} = \|R_n\|_\infty + 0.35\,\mathrm{RMS}(R_n) + 0.10\|R_{\mathrm{int}}\|_\infty + 0.05\,\mathrm{RMS}(R_{\mathrm{int}}). \tag{4.17}$$

A final coefficient-only L-BFGS step keeps α_Θ and the PHS centers fixed. The training procedure is:

1. initialize the PHS representation and the endpoint coordinate;
2. construct the training/reference Caputo and auto-convolution quadratures and the independent integral-residual quadrature from the current detached coordinate;
3. minimize the differential, integral, reference-level, and consistency-gap residuals with AdamW;
4. refresh the quadratures and enrich the collocation set from the normalized residual;
5. continue AdamW optimization on the enriched set;
6. start the deterministic L-BFGS restarts from the same selected Adam checkpoint;

7. select the final state by the held-out physics score (4.17);
8. perform independent high-order residual and exact-solution validation.

The exact solution is not used in training or model selection. It is used to construct the manufactured source and for independent operator and error checks.

For the two-dimensional Eq. (1.3), the endpoint coordinate remains one-dimensional:

$$u_\Theta(\rho, y) = zF_\Theta(z, y), \quad z = \rho^{\alpha_\Theta}. \tag{4.18}$$

The regular factor contains $m = 34$ trainable cubic PHS features in z with four Legendre modes in y, a tensor polynomial with powers $z^0, \dots, z^5$ and $P_0, \dots, P_3$ in the second variable, and a two-input tanh correction network with two hidden layers of 24 units each. The construction is anisotropic because the fractional endpoint singularity occurs only in the x-direction. For the double auto-convolution term, use

$$\xi = 1 + (x - 1)t, \quad \eta = yq. \tag{4.19}$$

The t-integration contains the weak factor $(1 - t)^{-\sigma}$ and is evaluated by the fitted Gauss–Jacobi rule, whereas the smooth q-integration is evaluated by Gauss–Legendre quadrature. At fixed y, the derivative-free Caputo and auto-convolution operators use training and higher-order reference discretizations, whereas the equivalent integral residual is evaluated with its separate fitted quadrature.

5. Consistency, stability, and error analysis

The equivalent integral equation and the contraction constant in Theorem 2.3 are used for the error analysis. The differential residual is kept as a numerical condition, while the integral residual is used in the stability estimate. We use Gauss–Jacobi approximation results [46–48], perturbation estimates for Volterra operators [5,6], and residual error estimates for physics-informed methods [49,50]. For one quadrature stage, the fitted exponents, nodes, and weights are fixed and the corresponding operator $\mathcal{T}_Q$ is analyzed. After a refresh, the same estimates apply to the new fixed operator.

After the singular factors are separated, the three nonlocal terms are approximated by weighted quadrature rules. Let

$$I_{a,b}[\phi] = \int_0^1 t^b(1 - t)^a \phi(t)\, \mathrm{d}t, \quad a, b > -1, \tag{5.1}$$

and let $Q_Q^{a,b}$ be the Q-point Gauss–Jacobi rule for this weight. The following estimate follows from Gaussian exactness and positivity of the Jacobi weights. Related Gauss–Jacobi error estimates and best polynomial approximation results are given in [46,48].

Lemma 5.1 (Gauss–Jacobi error through polynomial approximation). *Let $\phi \in C[0, 1]$ and $m_{a,b} = \mathrm{B}(b + 1, a + 1)$. Then*

$$|I_{a,b}[\phi] - Q_Q^{a,b}[\phi]| \le 2m_{a,b}\, E_{2Q-1}(\phi), \tag{5.2}$$

where $E_n(\phi) = \inf_{p \in \mathbb{P}_n} \|\phi - p\|_\infty$. Consequently, if $r \in \mathbb{N}$ and $\phi \in C^r[0, 1]$, then

$$|I_{a,b}[\phi] - Q_Q^{a,b}[\phi]| \le C_{a,b,r} Q^{-r} \|\phi\|_{C^r}, \tag{5.3}$$

whereas analyticity in a Bernstein ellipse of parameter $\varrho > 1$ gives

$$|I_{a,b}[\phi] - Q_Q^{a,b}[\phi]| \le C_{a,b,\varrho} \varrho^{-2Q}. \tag{5.4}$$

Proof. For any $p \in \mathbb{P}_{2Q-1}$, Gaussian exactness gives $I_{a,b}[p] = Q_Q^{a,b}[p]$. Since the Gauss–Jacobi weights are positive and sum to $m_{a,b}$,

$$|I_{a,b}[\phi] - Q_Q^{a,b}[\phi]| \le 2m_{a,b} \|\phi - p\|_\infty.$$

Taking the best polynomial approximation gives (5.2). The algebraic and analytic consequences follow from standard polynomial approximation estimates; see [46,48]. □

For Proposition 3.1, the weighted integrand is

$$\phi_C(q; \rho) = \frac{w(\rho) - w(\rho(1 - q))}{q}, \quad I_{0,-\mu}[\phi_C]. \tag{5.5}$$

For the weak Volterra term, the fitted rule in (3.2) has weight $t^{a_q}(1 - t)^{a_q - \sigma}$ and regular factor

$$\phi_V(t) = \frac{C(t)}{t^{a_q}(1 - t)^{a_q}}. \tag{5.6}$$

For the equivalent integral residual, (3.5) has weight $t^{b_q}(1 - t)^{\mu - 1}$ and regular factor

$$\phi_I(t; \rho) = \frac{H_\rho(t)}{t^{b_q}}. \tag{5.7}$$

The quadrature error depends on the smoothness of the factor that remains after the singular weight is removed. This is the product-integration principle used for weakly singular Volterra equations [12] and in weighted quadrature for fractional neural methods [25,30,36]. In two dimensions, Lemma 5.1 is used in the singular direction and Gauss–Legendre quadrature in the smooth direction. The uniform operator estimate used below is assumed for the trial class.

Let $\mathcal{T}_Q$ be a fixed numerical approximation of (2.11), including the fitted fractional-integral and Volterra quadratures. For $v \in B_R$, define

$$\eta_Q(v) = \|v - \mathcal{T}_Q v\|_\infty, \quad \varepsilon_Q(v) = \|\mathcal{T} v - \mathcal{T}_Q v\|_\infty. \tag{5.8}$$

The following bound follows from the contraction property of $\mathcal{T}$. Similar perturbation estimates are used for Volterra equations [5,6], fully discrete Caputo–Hadamard equations [16], and residual error estimates for PINNs [49,50].

Theorem 5.2 (Residual-to-solution error bound). *Under the assumptions of Theorem 2.3, let w be the exact fixed point and let $v \in B_R$. Then*

$$\|w - v\|_\infty \le \frac{\eta_Q(v) + \varepsilon_Q(v)}{1 - \kappa_h}. \tag{5.9}$$

Proof. Using $w = \mathcal{T} w$,

$$\|w - v\|_\infty \le \|\mathcal{T} w - \mathcal{T} v\|_\infty + \|\mathcal{T} v - \mathcal{T}_Q v\|_\infty + \|\mathcal{T}_Q v - v\|_\infty.$$

The contraction estimate (2.17) gives

$$(1 - \kappa_h)\|w - v\|_\infty \le \varepsilon_Q(v) + \eta_Q(v),$$

which proves (5.9). □

To compare the training and reference discretizations, let Q_{ref} denote a fixed higher-order rule and set

$$\delta_{Q,Q_{\mathrm{ref}}}(v) = \|\mathcal{T}_{Q_{\mathrm{ref}}} v - \mathcal{T}_Q v\|_\infty, \quad \varepsilon_{\mathrm{ref}}(v) = \|\mathcal{T} v - \mathcal{T}_{Q_{\mathrm{ref}}} v\|_\infty. \tag{5.10}$$

Corollary 5.3 (Error bound). *Under the assumptions of Theorem 5.2,*

$$\|w - v\|_\infty \le \frac{\eta_Q(v) + \delta_{Q,Q_{\mathrm{ref}}}(v) + \varepsilon_{\mathrm{ref}}(v)}{1 - \kappa_h}. \tag{5.11}$$

Proof. By the triangle inequality,

$$\begin{aligned} \varepsilon_Q(v) &= \|\mathcal{T} v - \mathcal{T}_Q v\|_\infty \\ &\le \|\mathcal{T} v - \mathcal{T}_{Q_{\mathrm{ref}}} v\|_\infty + \|\mathcal{T}_{Q_{\mathrm{ref}}} v - \mathcal{T}_Q v\|_\infty \\ &= \varepsilon_{\mathrm{ref}}(v) + \delta_{Q,Q_{\mathrm{ref}}}(v). \end{aligned}$$

Substituting this inequality into Theorem 5.2 yields (5.11). □

Corollary 5.3 concerns two discretizations of the complete fixed-point operator. In the numerical calculation, the integral residual has its own fitted quadrature, while the higher-order comparison is applied to the Caputo and auto-convolution terms. The sampled training–reference differences are therefore reported as numerical resolution checks and not as values of $\delta_{Q,Q_{\mathrm{ref}}}$ for the complete operator.

The quantity $\delta_{Q,Q_{\mathrm{ref}}}$ is a continuous sup-norm quantity. In the computations it is evaluated on a finite reference set and is reported as a

numerical estimate. Lemma 5.1 bounds $\varepsilon_{\mathrm{ref}}$ when the corresponding regular factors satisfy the required uniform smoothness assumptions.

Let $\mathcal{V}_m \subset B_R$ be a trial class and define

$$e_m = \inf_{v\in\mathcal{V}_m} \|w - v\|_\infty. \tag{5.12}$$

Assume a uniform quadrature consistency bound

$$\sup_{v\in\mathcal{V}_m} \|\mathcal{T}v - \mathcal{T}_Q v\|_\infty \le \epsilon_Q. \tag{5.13}$$

Suppose the optimizer returns $w_{m,Q} \in \mathcal{V}_m$ satisfying

$$\eta_Q(w_{m,Q}) \le \inf_{v\in\mathcal{V}_m} \eta_Q(v) + \epsilon_{\mathrm{opt}}. \tag{5.14}$$

Eqs. (5.13) and (5.14) are assumptions used in the estimate. Eq. (5.14) is not a consequence of AdamW or L-BFGS training. Similar error decompositions are considered in [49,50].

Theorem 5.4 (Conditional error decomposition). *Under the assumptions above,*

$$\|w - w_{m,Q}\|_\infty \le \frac{(1+\kappa_h)e_m + 2\epsilon_Q + \epsilon_{\mathrm{opt}}}{1-\kappa_h}. \tag{5.15}$$

Consequently, for a sequence of trial classes and fixed-stage quadrature rules such that

$$e_m \to 0, \quad \epsilon_Q \to 0, \quad \epsilon_{\mathrm{opt}} \to 0, \tag{5.16}$$

one has $\|w - w_{m,Q}\|_\infty \to 0$ *in the local contractive regime.*

Proof. Let $p_m \in \mathcal{V}_m$ satisfy $\|w - p_m\|_\infty \le e_m + \varepsilon$ for arbitrary $\varepsilon > 0$. Then

$$\begin{aligned}\eta_Q(p_m) &\le \|p_m - w\|_\infty + \|\mathcal{T}w - \mathcal{T}p_m\|_\infty + \|\mathcal{T}p_m - \mathcal{T}_Q p_m\|_\infty \\ &\le (1+\kappa_h)(e_m+\varepsilon) + \epsilon_Q.\end{aligned}$$

Using (5.14) and then Theorem 5.2,

$$\|w - w_{m,Q}\|_\infty \le \frac{(1+\kappa_h)(e_m+\varepsilon) + 2\epsilon_Q + \epsilon_{\mathrm{opt}}}{1-\kappa_h}.$$

Letting $\varepsilon \downarrow 0$ proves (5.15). The convergence statement follows immediately from the stated hypotheses. □

Remark 5.5. Theorem 5.4 is conditional on the approximation, quadrature, and optimization errors. The condition $e_m \to 0$ is an approximation assumption for the trial classes. Polyharmonic-spline interpolation theory gives error estimates under smoothness and interpolation assumptions [42–45], but these results do not prove $e_m \to 0$ for the trainable PHS–neural class used here. The uniform quadrature condition (5.13) is also an assumption on the fitted regular factors. The condition $\epsilon_{\mathrm{opt}} \to 0$ is an optimization assumption. Multi-start optimization and held-out selection are used in the numerical procedure to reduce the observed dependence on a single optimization run.

6. Numerical experiments

All calculations are performed in double precision. Five independent CPU runs are used for the principal one-dimensional results. Each run uses the same numerical setting. The run seed changes the initialization of the trainable parameters and the stochastic sampling and enrichment steps. The same repeated-run protocol is used for the two-dimensional verification. The one-dimensional runs reported below were assessed on the same fixed validation set

$$\begin{aligned}\mathcal{V} = &\left\{ 10^{-7} + (1-10^{-7})\frac{j}{900} \;:\; 0 \le j \le 900 \right\} \\ &\cup \left\{ 10^{-9} + (1-10^{-9})\left(\frac{k}{600}\right)^{3.5} \;:\; 0 \le k \le 600 \right\},\end{aligned} \tag{6.1}$$

after duplicate nodes were removed. The reported relative validation error is

$$E_{\mathcal{V}} = \left(\frac{\sum_{\rho_i\in\mathcal{V}} |w_\Theta(\rho_i) - w_{\mathrm{ex}}(\rho_i)|^2}{\sum_{\rho_i\in\mathcal{V}} |w_{\mathrm{ex}}(\rho_i)|^2} \right)^{1/2}. \tag{6.2}$$

The set $\mathcal{V}$ is finite, nonuniform, and refined near the endpoint. Hence $E_{\mathcal{V}}$ is reported as a discrete relative validation error and not as the continuous $L^2(0,1)$ norm. The same set and formula are used for all one-dimensional results. Reported standard deviations are sample standard deviations, and the success count in Table 4 uses $E_{\mathcal{V}} < 10^{-4}$.

The five seeds are 1234, 42, 2025, 2026, and 7777. For these runs, the mean, sample standard deviation, and individual errors are reported. A bootstrap calculation with 200,000 samples gives the 95% interval $[2.0672\times10^{-5},\ 4.8488\times10^{-5}]$ for the mean $E_{\mathcal{V}}$. Because the sample contains five runs, this interval is reported together with the individual errors. The value 10^{-4} is used only for the numerical success count; it is about three times the mean validation error and is above the exact-solution residual floor 1.02×10^{-7}.

The principal calculation uses 1400 persistent training points, 420 held-out points, batch size 220, and a cap of 1900 points after enrichment. The two AdamW stages use 1600 and 850 iterations; the learning rates are 7.0×10^{-4} for the representation parameters and 1.25×10^{-4} for the endpoint-coordinate and center parameters. The three deterministic L-BFGS restarts use 620-point objectives with nominal iteration limits 320, 320, and 240, followed by a coefficient-only polish with a nominal limit of 140 iterations. The training Caputo and Volterra rules use 40 nodes, the higher-order reference rules use 56 nodes, and the independent integral residual uses $Q_I = 14$ nodes. Independent final residuals are evaluated with 220-node Caputo and Volterra rules. Training, reference, source, and final-validation quadratures use different orders. For the manufactured examples, the exact solution defines the source and its nonlocal term is evaluated by a separate high-order quadrature. The exact-solution residual is then used to measure the numerical source and operator error.

The principal one-dimensional example is used to examine the endpoint behavior and the nonlocal operators. We use the standard Mittag–Leffler notation $E_{\alpha,\beta}(z) = \sum_{k=0}^{\infty} z^k/\Gamma(\alpha k+\beta)$ and $E_\alpha = E_{\alpha,1}$. For the principal benchmark, the general Eq. (1.1) is specialized to

$$\mu = 0.6, \quad \beta = 0.25, \quad \sigma = 0.4,$$

$$a(x) \equiv a_0 = -\frac{1}{2}, \quad \psi(x,s) = 1 + s, \tag{6.3}$$

and the manufactured solution

$$v_{\mathrm{ex}}(x) = E_{1/2}\left(\sqrt{\log x}\right) - 1. \tag{6.4}$$

The source g is obtained by substituting this exact solution together with $a(x) \equiv -1/2$ and $\psi(x,s) = 1+s$ into (1.1). In logarithmic coordinates,

$$w_{\mathrm{ex}}(\rho) = e^{\rho}\,\mathrm{erfc}(-\sqrt{\rho}) - 1 = \frac{2}{\sqrt{\pi}}\rho^{1/2} + O(\rho), \quad \rho \downarrow 0. \tag{6.5}$$

The exact solution is used to construct the source and for independent operator and error checks. The five-run results for the proposed approximation are reported together with the controlled approximation-family comparison in Table 4. Fig. 2 shows the relative validation errors over five runs for each of the three approximation families.

For this example, $\gamma = 0.15$. On $0 \le \rho \le 0.01$, a 500-point logarithmically refined calculation gives $\max|g(\rho)|/\rho^{0.15} = 1.1823$. With $G = 1.20$, $A_h = 0.5$, $M_h = e^h(1+e^h)$, and $R = 0.19$, Eqs. (2.16) and (2.17) give $0.1722 < R$ and $\kappa_h = 0.1519 < 1$. Also, $\max_{[0,h]}|w_{\mathrm{ex}}| = 0.1237 < R$. These values verify the contraction conditions on this interval. No contraction value is reported for the full interval $[0,1]$.

Table 1 reports the exact-solution errors for increasing quadrature order.

All three errors in Table 1 decrease as Q increases. The source quadrature is compared with an order-880 calculation on the endpoint-refined validation set. The maximum differences are 2.23×10^{-4}, 2.52×10^{-5}, 2.79×10^{-6}, 7.67×10^{-7}, 1.10×10^{-7}, and 1.08×10^{-8} for source orders 20, 40, 80, 120, 220, and 440. The decreasing values give an independent numerical check of the source and validation quadratures.

For the approximation-family comparison, the transformation, nonlocal operators, adaptive points, reference discretization, integral residual, held-out selection, and optimization procedure are kept fixed. Only

Table 1
Exact-solution quadrature errors for the one-dimensional example. The source rule has order at least 440, and Q is used for the Caputo and auto-convolution validation rules. The training, reference, and final-validation orders are 40, 56, and 220.

Q	Caputo absolute error	Volterra order gap	equation-residual floor
10	1.1721×10^{-4}	1.8932×10^{-3}	1.9267×10^{-3}
20	1.5155×10^{-5}	2.2268×10^{-4}	2.2703×10^{-4}
40	1.9250×10^{-6}	2.5194×10^{-5}	2.5748×10^{-5}
80	2.4241×10^{-7}	2.7833×10^{-6}	2.8530×10^{-6}
120	7.1593×10^{-8}	7.5619×10^{-7}	7.7690×10^{-7}
220	1.1634×10^{-8}	9.8893×10^{-8}	1.0227×10^{-7}

Table 2
Ablation results from three runs for the one-dimensional example. Values are mean ± sample standard deviation of E_y for the reduced numerical configuration.

Configuration	Mean ± SD E_y
Full reduced configuration	$3.979 \times 10^{-3} \pm 6.623 \times 10^{-4}$
Without reference-level residual	$2.276 \times 10^{-3} \pm 2.915 \times 10^{-4}$
Without integral residual	$3.595 \times 10^{-3} \pm 1.682 \times 10^{-3}$
Without adaptive enrichment	$3.532 \times 10^{-3} \pm 1.148 \times 10^{-3}$
Fixed endpoint coordinate 0.5	$2.688 \times 10^{-3} \pm 1.516 \times 10^{-4}$
Without cubic-PHS component	$2.520 \times 10^{-3} \pm 5.624 \times 10^{-4}$
Fixed quadrature fitting exponent	$4.209 \times 10^{-3} \pm 3.767 \times 10^{-4}$
Without endpoint fitting in Jacobi rules	$3.573 \times 10^{-3} \pm 1.764 \times 10^{-3}$
One L-BFGS restart	$3.979 \times 10^{-3} \pm 6.623 \times 10^{-4}$

Table 3
Parameter sensitivity results for the one-dimensional example under the reduced numerical setting.

μ	β	σ	E_y	learned α_Θ
0.60	0.25	0.40	3.351×10^{-3}	0.5015
0.50	0.25	0.40	9.274×10^{-2}	0.4215
0.70	0.25	0.40	3.733×10^{-1}	0.6266
0.60	0.10	0.40	3.278×10^{-3}	0.4989
0.60	0.35	0.40	6.105×10^{-3}	0.5095
0.60	0.25	0.20	1.400×10^{-3}	0.5008
0.60	0.25	0.60	3.238×10^{-3}	0.5013
0.60	0.25	0.70	5.764×10^{-3}	0.5004
0.60	0.25	0.75	5.338×10^{-3}	0.5016

the trial family is changed. The tanh model has three hidden layers of width 24 and 1274 parameters. The Gaussian RBF model has 150 trainable centers and widths and 453 parameters.

The tanh network uses Xavier initialization and the same five seeds, collocation points, adaptive enrichment, quadrature rules, AdamW/L-BFGS steps, and held-out selection. The Gaussian RBF model uses 150 trainable centers and widths with initial width 0.08 under the same setting. Since the parameter counts are different, Table 4 reports only the common-setting comparison.

Table 2 reports three-run tests for the reduced numerical setting. One component is removed in each case. The errors vary with the component and do not change monotonically. The fixed value $\alpha_\Theta = 0.5$ is the known endpoint power of this manufactured example and is used only for comparison. Without endpoint fitting, the physical factor $(1-t)^{-\sigma}$ is used and the learned endpoint powers are omitted from the auto-convolution and integral Jacobi rules. The one-restart case uses the first L-BFGS run only. In the fixed-quadrature case, the fitting exponent remains at its initial value while the endpoint coordinate is trainable.

For seed 2025, the initial values $\alpha_\Theta(0) = 0.25$, 0.43, and 0.70 give $E_y = 2.4930 \times 10^{-2}$, 4.6707×10^{-3}, and 1.5830×10^{-2}. The learned values are 0.28718, 0.50121, and 0.51032. The three errors depend on the initial endpoint coordinate.

The largest errors in Table 3 occur for $\mu = 0.50$ and $\mu = 0.70$, where $E_y = 9.274 \times 10^{-2}$ and 3.733×10^{-1}. For $\beta = 0.10$ and 0.35, the source power $\gamma = \nu + \beta - \mu$ changes while the exact endpoint power remains $\nu = 0.5$. The cases $\sigma = 0.70$ and 0.75 give errors 5.764×10^{-3} and 5.338×10^{-3}. With $a = -0.1$, the error is 5.090×10^{-3} and the learned value is $\alpha_\Theta = 0.5016$. Endpoint-coordinate values for the five production runs are not available. These results describe sensitivity for the listed cases and are not a uniform error result for the full parameter range.

In the common setting, the tanh mean error is about 4.19 times the SI-OC-PHS-fPINN mean error, and the Gaussian mean error is about 178 times larger. These ratios refer only to the three implementations in Table 4; they are not comparisons with published Spectral-fPINN or PINNIES results.

The two-dimensional test uses the same endpoint and operator construction for the weakly singular auto-convolution model. The dimensional extension specializes (1.3) to

$$\mu = 0.6, \quad \beta = 0.25, \quad \sigma = 0.4, \qquad a(x,y) \equiv -\frac{1}{2}, \quad \Psi(x,y,\xi,\eta) = 1 + \xi + \eta. \tag{6.6}$$

The manufactured solution is

$$u_{\mathrm{ex}}(x,y) = e^{y}\left[E_{0.35}\left((\log x)^{0.35}\right) - 1\right] + y(1-y)(\log x)^{1.35}, \tag{6.7}$$

with

$${}^{\mathrm{CH}}_{1}D_{x}^{0.6}u_{\mathrm{ex}}(x,y) = e^{y}(\log x)^{-0.25}E_{0.35,0.75}\left((\log x)^{0.35}\right) + y(1-y)\frac{\Gamma(2.35)}{\Gamma(1.75)}(\log x)^{0.75}. \tag{6.8}$$

Near the third-kind endpoint,

$$u_{\mathrm{ex}}(1+t,y) \sim \frac{e^{y}}{\Gamma(1.35)}t^{0.35}, \quad t \downarrow 0. \tag{6.9}$$

The source G is manufactured from (1.3). Five independent double-precision CPU runs use the same two-dimensional architecture and numerical settings. The two-dimensional configuration uses 900 persistent training points, 240 held-out points, batch size 96, and AdamW stages of 800 and 420 iterations. Three deterministic L-BFGS restarts are then launched from the selected Adam checkpoint on 200-point objectives with nominal iteration limits 100, 100, and 150, followed by a coefficient-only polish with a nominal limit of 100 iterations. The training quadrature orders are $Q_C = 20$ and $(Q_{V,x}, Q_{V,y}) = (16, 8)$; the higher-order reference orders are $Q_C^{\mathrm{ref}} = 28$ and $(Q_{V,x}^{\mathrm{ref}}, Q_{V,y}^{\mathrm{ref}}) = (22, 11)$; the independent integral residual uses $Q_I = 6$; and the independent final validation uses $Q_C^{\mathrm{val}} = 42$ and $(Q_{V,x}^{\mathrm{val}}, Q_{V,y}^{\mathrm{val}}) = (36, 18)$. Residual enrichment evaluates a pool of 1400 candidates, adds 280 points subject to a training cap of 1100, and applies two short residual-gradient moves to 80 selected candidates. For the two-dimensional error, define the uniform tensor grid

$$\mathcal{G} = \{(\rho_i, y_j) : \rho_i = i/120,\ 0 \le i \le 120,\ y_j = j/50,\ 0 \le j \le 50\}, \tag{6.10}$$

and the relative grid error

$$E_{\mathcal{G}} = \left(\frac{\sum_{(\rho_i,y_j)\in\mathcal{G}} |u_\Theta(\rho_i,y_j) - u_{\mathrm{ex}}(\rho_i,y_j)|^2}{\sum_{(\rho_i,y_j)\in\mathcal{G}} |u_{\mathrm{ex}}(\rho_i,y_j)|^2}\right)^{1/2}. \tag{6.11}$$

The endpoint diagnostic $\bar{\nu}_{\mathrm{eff}}$ is the arithmetic mean of slopes fitted at $y = 0$, 0.25, 0.50, 0.75, and 1 using 28 logarithmically spaced values of ρ on $[10^{-6}, 10^{-3}]$. Table 5 reports the results of five independent runs for the two-dimensional test. Fig. 3 shows the exact solution, computed solution, and corresponding error for the two-dimensional test.

For seed 2025, three calculations use increasing numbers of training points $100 \to 360 \to 900$, PHS centers $12 \to 24 \to 34$, correction-network widths $10 \to 18 \to 24$, and training quadrature orders $(6,6,4) \to (12,10,6) \to (20,16,8)$. The relative grid error decreases $0.11684 \to 0.05087 \to 0.02686$.

For validation orders $(Q_C, Q_{V,x}, Q_{V,y}) = (10,8,4), (20,16,8), (28,22,11), (42,36,18), (56,48,24)$, the normalized residuals are 2.24×10^{-4}, 3.5×10^{-5}, 1.4×10^{-5}, 4.0×10^{-6}, and 2.0×10^{-6}. The auto-convolution order gap decreases from 2.983×10^{-2} to 2.37×10^{-4}.

Table 4
Five-run controlled comparison on the principal benchmark. All rows use the common validation set (6.1); only the approximation family is changed. Reported central values are arithmetic means ± sample standard deviations.

Method	Params	Mean±SD E_V	Median	Worst	$< 10^{-4}$
SI-OC-PHS-fPINN	476	$3.458 \times 10^{-5} \pm 1.840 \times 10^{-5}$	2.416×10^{-5}	5.559×10^{-5}	5/5
Tanh MLP, common setting	1274	$1.449 \times 10^{-4} \pm 7.254 \times 10^{-5}$	1.299×10^{-4}	2.624×10^{-4}	1/5
Gaussian RBF, common setting	453	$6.171 \times 10^{-3} \pm 1.720 \times 10^{-3}$	6.126×10^{-3}	8.469×10^{-3}	0/5

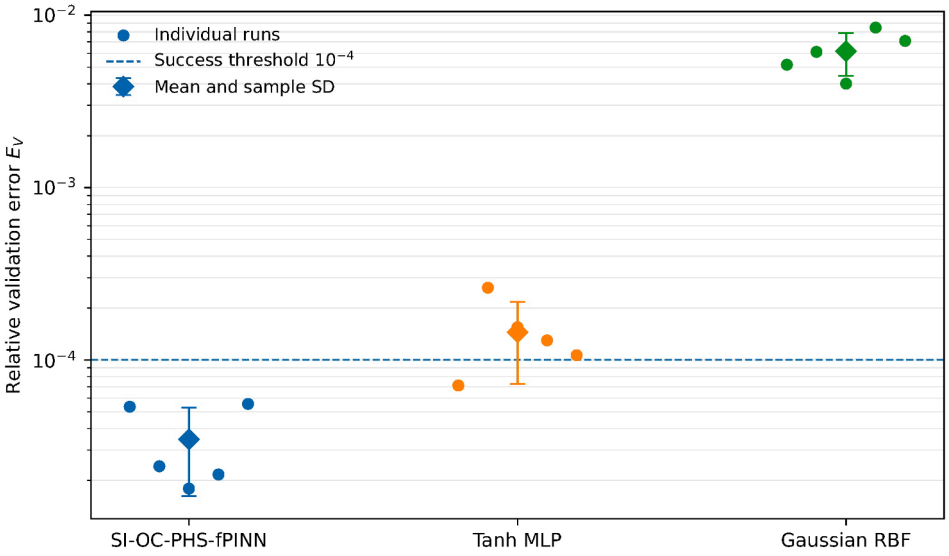


Fig. 2. Five-run relative validation errors E_V for the three controlled approximation families under the same logarithmic transformation, nonlocal operators, adaptive enrichment, higher-order reference discretizations and the independent integral residual, and multi-start optimization. Markers show individual runs; diamonds with error bars show the arithmetic mean ± sample standard deviation. The horizontal dashed line marks 10^{-4}.

Table 5
Five-run two-dimensional verification. E_G is the relative error on the 121×51 uniform tensor grid in (ρ, y).

Run	E_G	$\|R_n\|_\infty$	α_Θ	$\bar{\nu}_{\mathrm{eff}}$
1	2.7260×10^{-2}	6.6123×10^{-3}	0.349615	0.360171
2	2.4095×10^{-2}	9.8025×10^{-3}	0.346882	0.358077
3	2.6862×10^{-2}	1.1254×10^{-2}	0.354603	0.364191
4	2.9283×10^{-2}	1.0932×10^{-2}	0.363642	0.368615
5	2.2449×10^{-2}	6.4505×10^{-3}	0.349989	0.360543
Mean±SD	$2.5990 \times 10^{-2} \pm 2.7085 \times 10^{-3}$	$9.0102 \times 10^{-3} \pm 2.3268 \times 10^{-3}$	0.352946 ± 0.006591	0.362320 ± 0.004151

A nonseparable test uses the smooth coefficients,

$$a(x, y) = -\frac{1}{2}\left[1 + 0.15\, y(x-1)\right], \Psi(x, y, \xi, \eta) = 1 + 0.15xy + 0.25\xi\eta + 0.10y\xi, \tag{6.12}$$

and the source is computed from Eq. (1.3). With seed 2025 and the medium numerical setting, $E_G = 1.6404 \times 10^{-2}$, $\|R_n\|_\infty = 2.2687 \times 10^{-2}$, $\alpha_\Theta = 0.37464$, and the exact-solution equation-residual floor is 8.28×10^{-6}.

7. Discussion

For the nonlocal discretization, Lemma 5.1 bounds the quadrature error through approximation of the regular factor after the singular weight is included in the Jacobi rule. Corollary 5.3 compares two discretizations of the integral fixed-point operator. In the computation, the training–reference comparison is applied to the Caputo and auto-convolution terms, while the integral residual uses its own fitted rule.

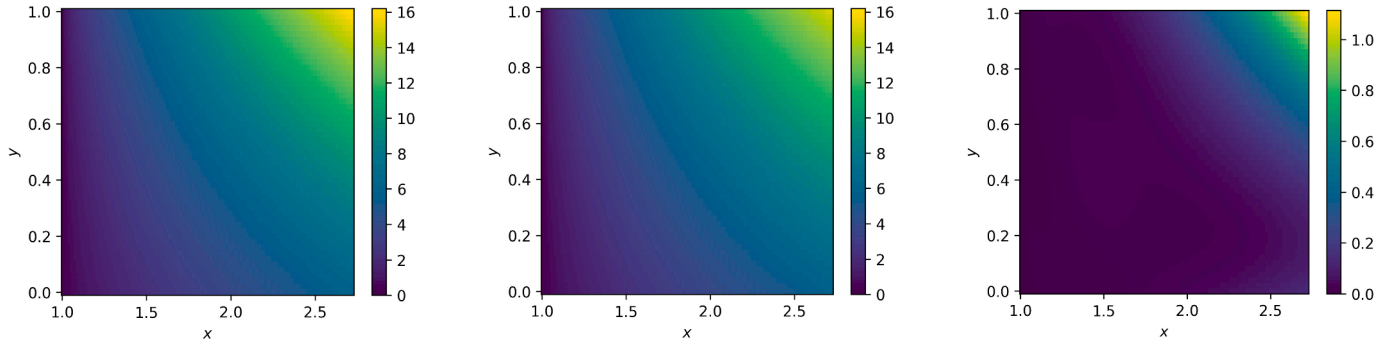
Optimization enters Theorem 5.4 through ϵ_{opt}. The theorem does not require the numerical optimizer to find a global minimizer. Several L-BFGS runs are compared on held-out residuals so that the final state is not selected from one trajectory only.

The numerical comparison in Section 6 uses a common setting and changes only the approximation family. Spectral-fPINN and PINNIES are discussed as related methods, but no error values from those papers are included in Table 4. The method studied here uses a learned endpoint coordinate, a derivative-free Caputo–Hadamard residual, fitted quadrature for the weak kernel, an integral residual, and a higher-order reference evaluation for the third-kind auto-convolution equation.

The bounds are local. For the principal example, Theorem 2.3 is verified on $0 \le \rho \le 0.01$ with $R = 0.19$ and $\kappa_h = 0.1519 < 1$. No contraction constant is reported for the full interval $[0, 1]$; the results on that interval are numerical.

For the local fixed-point estimate, continuity in y is sufficient because no singular weight occurs in the y-integration. In the numerical calculation, y is represented by Legendre modes and integrated by Gauss–Legendre quadrature, and the manufactured solution is smooth in this variable. A singularity in the y direction is not included in these calculations.

In the five two-dimensional runs, the endpoint scale is recovered close to the theoretical value, while the grid error remains at the percent level. The one-dimensional accuracy and the two-dimensional end-



(a) Exact solution (b) Computed solution (c) Error

Fig. 3. Numerical results.

point result are therefore reported separately. The calculations do not establish dimension-independent accuracy.

8. Conclusion

A singularity-inferred operator-consistent fractional PINN has been developed for third-kind Caputo–Hadamard auto-convolution Volterra equations with weakly singular kernels. The logarithmic transformation converts the Caputo–Hadamard derivative to a Caputo derivative, and a derivative-free representation places solution values inside the fractional quadrature. The learned endpoint coordinate is shared by the PHS trial function and the weakly singular quadrature weights, while an independently discretized equivalent fractional-integral equation and higher-order reference evaluations of the Caputo and auto-convolution operators provide complementary constraints.

For algebraic source behavior, the analysis gives local existence and uniqueness, the endpoint law, residual stability, and an error bound containing approximation, quadrature, and optimization terms. The reference residual is a numerical finite-set check and is not used as a continuous operator norm. The error theorem is conditional on the stated approximation, quadrature, and optimization assumptions.

The five one-dimensional runs give low validation errors, and the tanh and Gaussian-RBF models give larger mean errors in the same numerical setting. In two dimensions, the endpoint scale is recovered in all five runs, while the grid error is larger. The numerical tests use manufactured solutions on structured domains. Non-manufactured problems and more accurate two-dimensional calculations are left for further work.

CRediT authorship contribution statement

Essaid Aourir: Writing – original draft, Visualization, Software, Methodology, Investigation; **Hojatollah Laeli Dastjerdi:** Writing – review & editing, Visualization, Validation, Software, Methodology, Investigation; **Kamal Shah:** Visualization, Validation, Formal analysis, Conceptualization; **Thabet Abdeljawad:** Visualization, Validation, Formal analysis, Conceptualization.

Funding

The authors did not receive support from any organization for the submitted work.

Reproducibility and code availability

The source code is not supplied with the article. The numerical settings reported above include the transformation, trial function, random seeds, quadrature orders, quadrature refresh, adaptive points, stopping limits, held-out selection, and source construction. The manufactured source is obtained from the prescribed exact solution. Its Caputo–Hadamard term is evaluated analytically and its auto-convolution term by an independent high-order quadrature. In one dimension, source orders up to 440 are compared with order 880. In two dimensions, the source check uses separate Gauss–Jacobi/Gauss–Legendre orders $(48, 24)$. The source rules are separate from the training and final-validation rules.

In one dimension, α_Θ is initialized at 0.43 and restricted to $[0.15, 0.90]$. The cubic-PHS weights are initialized from $N(0, 0.015^2)$, the center-gap logits are zero, and the degree-six polynomial starts from $(1.05, 0.10, 0, \ldots, 0)$. The correction network uses Xavier normal weights with gain 0.45 and zero biases. All production calculations use double precision on CPU. The two-dimensional calculations use Python 3.12, NumPy 2.1.3, SciPy 1.16.1, PyTorch 2.8.0+cpu, and Matplotlib 3.10.5 on Windows 10. The five computation times are about 134–425 s, with median about 335 s. The processor model is not available. The seeds are 1234, 42, 2025, 2026, and 7777.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The authors would like to thank Prince Sultan University for support through the TAS Research Laboratory.

Data availability

No data was used for the research described in the article.

References

[1] J. Janno, L. von Wolfersdorf, Integro-differential equations of first order with auto-convolution integral, J. Integral Equ. Appl. 21 (2009) 39–75.
[2] R. Zhang, H. Liang, H. Brunner, Analysis of collocation methods for generalized auto-convolution Volterra integral equations, SIAM J. Numer. Anal. 54 (2016) 899–920.
[3] Y. Li, Z. Yang, C. Zhang, Theoretical and numerical analysis of third-kind auto-convolution Volterra integral equations, Comput. Appl. Math. 38 (2019) 170.
[4] Y. Li, Z. Yang, H. Liang, Analysis of collocation methods for a class of third-kind auto-convolution Volterra integral equations, Math. Comput. Simul. 199 (2022) 341–358.
[5] G. Gripenberg, S.-O. Londen, O. Staffans, Volterra Integral and Functional Equations, Cambridge University Press, Cambridge, 1990.
[6] H. Brunner, Collocation Methods for Volterra Integral and Related Functional Differential Equations, Cambridge University Press, Cambridge, 2004.
[7] S.S. Allaei, Z.-W. Yang, H. Brunner, Collocation methods for third-kind VIEs, IMA J. Numer. Anal. 37 (2017) 1104–1124.
[8] X. Ma, C. Huang, Recovery of high order accuracy in spectral collocation method for linear Volterra integral equations of the third-kind with non-smooth solutions, J. Comput. Appl. Math. 392 (2021) 113458.
[9] Y. Guo, X. Xu, Z. Wang, Z. Wang, An hp-version Legendre collocation method for the third-kind VIEs with nonlinear vanishing delay, Math. Comput. Simul. 223 (2024) 338–350.

[10] H. Liang, H. Brunner, The convergence of collocation solutions in continuous piecewise polynomial spaces for weakly singular Volterra integral equations, SIAM J. Numer. Anal. 57 (2019) 1875–1896.
[11] H. Liang, M. Stynes, A general collocation analysis for weakly singular Volterra integral equations with variable exponent, IMA J. Numer. Anal. 44 (2024) 2725–2751.
[12] R. Katani, S. McKee, A product integration method for nonlinear second kind Volterra integral equations with a weakly singular kernel (with application to fractional differential equations), Appl. Math. Lett. 163 (2025) 109403.
[13] A.A. Kilbas, H.M. Srivastava, J.J. Trujillo, Theory and applications of fractional differential equations, vol. 204 of *North-Holland Mathematics Studies*, Elsevier, 2006.
[14] F. Jarad, T. Abdeljawad, D. Baleanu, Caputo-type modification of the Hadamard fractional derivatives, Adv. Differ. Equ. 2012 (2012) 142.
[15] T. Baghban, M.H. Heydari, M. Bayram, M.A. Zaky, A reliable strategy for a category of third-kind nonlinear fractional integro-differential equations, Chaos Solit. Fractals 200 (2025) 117022.
[16] E. Aourir, H.L. Dastjerdi, M.H. Heydari, K. Shah, T. Abdeljawad, Fully discrete RBF collocation method for linear and nonlinear one- and two-dimensional third-kind Caputo–Hadamard fractional delay integro-differential equations with nonlinear vanishing delays, Commun. Nonlinear Sci. Numer. Simul. 161 (2026) 110134.
[17] E. Aourir, H.L. Dastjerdi, On the numerical treatments of classes of third-kind auto-convolution VIEs via moving least squares collocation scheme, Eng. Comput. 42 (2025) 749–779.
[18] E. Aourir, N. Izem, H.L. Dastjerdi, Numerical scheme for solving third-kind auto-convolution VIEs using meshless collocation method, Appl. Anal. 104 (2025) 3147–3163.
[19] E. Aourir, H.L. Dastjerdi, M. Oudani, K. Shah, T. Abdeljawad, Numerical technique based on Bernstein polynomials approach for solving auto-convolution VIEs and the initial value problem of auto-convolution VIDEs, J. Appl. Math. Comput. 71 (2025) 4697–4727.
[20] M. Raissi, P. Perdikaris, G.E. Karniadakis, Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, J. Comput. Phys. 378 (2019) 686–707.
[21] S. Cuomo, V. Schiano di Cola, F. Giampaolo, G. Rozza, M. Raissi, F. Piccialli, Scientific machine learning through physics-informed neural networks: where we are and what's next, J. Sci. Comput. 92 (2022) 88.
[22] G. Pang, L. Lu, G.E. Karniadakis, fPINNs: fractional physics-informed neural networks, SIAM J. Sci. Comput. 41 (2019) A2603–A2626.
[23] L. Guo, H. Wu, X. Yu, T. Zhou, Monte Carlo fPINNs: deep learning method for forward and inverse problems involving high dimensional fractional partial differential equations, Comput. Methods Appl. Mech. Engrg. 400 (2022) 115523.
[24] S. Wang, G.E. Karniadakis, GMC-PINNs: a new general Monte Carlo PINNs method for solving fractional partial differential equations on irregular domains, Comput. Methods Appl. Mech. Engrg. 429 (2024) 117189.
[25] Z. Hu, K. Kawaguchi, Z. Zhang, G.E. Karniadakis, Tackling the curse of dimensionality in fractional and tempered fractional PDEs with physics-informed neural networks, Comput. Methods Appl. Mech. Engrg. 432 (2024) 117448.
[26] C. Wu, M. Zhu, Q. Tan, Y. Kartha, L. Lu, A comprehensive study of non-adaptive and residual-based adaptive sampling for physics-informed neural networks, Comput. Methods Appl. Mech. Engrg. 403 (2023) 115671.
[27] W. Ouyang, M. Zhu, W. Xiong, S.-W. Liu, L. Lu, RAMS: residual-based adversarial-gradient moving sample method for scientific machine learning in solving partial differential equations, Adv. Intell. Discov. 2 (2026) e202500214.
[28] J. Zhang, Y. Zhao, Y. Tang, Adaptive loss weighting auxiliary output fPINNs for solving fractional partial integro-differential equations, Phys. D 460 (2024) 134066.
[29] T. Zhang, D. Zhang, S. Shi, Z. Guo, Spectral-fPINNs: spectral method based fractional physics-informed neural networks for solving fractional partial differential equations, Nonlinear Dyn. 113 (2025) 12565–12588.
[30] Z. Lin, Q. Ma, H. Xie, X. Yin, Solving time-fractional partial integro-differential equations using tensor neural network, SIAM J. Sci. Comput. 48 (2026) C164–C189.
[31] J. Bai, G.-R. Liu, A. Gupta, L. Alzubaidi, X.-Q. Feng, Y. Gu, Physics-informed radial basis network (PIRBN): a local approximating neural network for solving nonlinear partial differential equations, Comput. Methods Appl. Mech. Engrg. 415 (2023) 116290.
[32] M. Mohammadi, R. Mokhtari, M. Ramezani, RBF-fPINNs: radial basis function-enhanced fractional physics-informed neural networks, Eng. Comput. 42 (2026) 42.
[33] X. Li, On simultaneous approximations by radial basis function neural networks, Appl. Math. Comput. 95 (1998) 75–89.
[34] A.A. Aghaei, M.M. Moghaddam, K. Parand, PINNIES: an efficient physics-informed neural network framework for integral operator problems, Comput. Phys. Commun. 327 (2026) 110255.
[35] R. Gao, L. Ju, D. Li, Kernel-based PINNs for time fractional differential equations and Volterra integral-differential equations, Phys. D 490 (2026) 135174.
[36] Z. Zhang, J. Li, Transformed diffusion-wave fPINNs: enhancing computational efficiency for time-fractional diffusion-wave equations, J. Comput. Phys. (2026) 115218. https://doi.org/10.1016/j.jcp.2026.115218
[37] H. Li, Z. Tan, DSC-PINN: a decomposed singularity-capturing PINN for time-fractional PDEs involving initial singularities, J. Comput. Phys. 561 (2026) 114965.
[38] H. Li, Z. Tan, A novel singularity- and discontinuity-capturing PINN for time-fractional diffusion equations involving initial singularities and interfaces on complex curved surfaces, Appl. Math. Comput. 519 (2026) 129917.
[39] Y. Kumar, R.K. Vats, A.K. Nain, S. Yadav, Robust deep learning framework using Hermite interpolation for time-dependent Caputo fractional differential equations, Math. Methods Appl. Sci. 49 (7) (2026) 6398–6427. https://doi.org/10.1002/mma.70395
[40] Y. Kumar, R.K. Vats, A deep Kolmogorov–Arnold network framework for solving time-fractional partial differential equations, Appl. Soft Comput. 188 (2026) 114379. https://doi.org/10.1016/j.asoc.2025.114379
[41] Y. Kumar, R.K. Vats, Deep wavelet neural network with lateral connections for solving fractional differential equations arising in oil pollution dynamics, Int. J. Comput. Math. 2026 (2026) 1–43. https://doi.org/10.1080/00207160.2026.2663999
[42] S.J. Hales, J. Levesley, Error estimates for multilevel approximation using polyharmonic splines, Numer. Algor. 30 (2002) 1–10.
[43] R.K. Beatson, H.-Q. Bui, J. Levesley, Embeddings of Beppo–Levi spaces in Hölder–Zygmund spaces, and a new method for radial basis function interpolation error estimates, J. Approx. Theory 137 (2) (2005) 166–178.
[44] H. Wendland, Scattered Data Approximation, Cambridge University Press, Cambridge, 2005.
[45] G.E. Fasshauer, Meshfree Approximation Methods with MATLAB, World Scientific, Singapore, 2007.
[46] W. Gautschi, Orthogonal Polynomials: Computation and Approximation, Oxford University Press, Oxford, 2004.
[47] G.H. Golub, J.H. Welsch, Calculation of Gauss quadrature rules, Math. Comp. 23 (1969) 221–230.
[48] W. Themistoclakis, Some error bounds for Gauss–Jacobi quadrature rules, Appl. Numer. Math. 116 (2017) 286–293.
[49] S. Mishra, R. Molinaro, Estimates on the generalization error of physics-informed neural networks for approximating PDEs, IMA J. Numer. Anal. 43 (2023) 1–43.
[50] T. De Ryck, A.D. Jagtap, S. Mishra, Error estimates for physics-informed neural networks approximating the Navier–Stokes equations, IMA J. Numer. Anal. 44 (2024) 83–119.