



NUMERICAL Solution of Fractional Reaction-Diffusion Equations using an Advanced Physics-Informed Neural Network

Maryam Mohammadi, Reza Mokhtari*, and Mohadese Ramezani

Department of Mathematical Sciences, Isfahan University of Technology, Isfahan 8415683111, Iran.

Abstract

This paper introduces an innovative approach featuring an advanced Physics-Informed Neural Network (PINN) that effectively tackles the challenges associated with fractional reaction-diffusion problems. These problems often pose difficulties for traditional numerical methods, especially in high-dimensional spaces or complex geometries. By employing a suitable auxiliary function, the approximate solution automatically satisfies the exact boundary conditions, further enhancing the method's efficiency and accuracy. Additionally, the proposed model can handle weak singularities in the solution, which are common in fractional models, making it particularly well-suited for more challenging cases. We conducted numerical experiments to demonstrate the effectiveness of the proposed framework. The results indicate that this framework significantly improves the performance of radial basis function neural networks, making them better suited for handling complex fractional models across different geometric configurations.

Keywords. Fractional PINN, Fractional reaction-diffusion equations, Weak singular solution, Auxiliary function.

2010 Mathematics Subject Classification. 65M22, 35R11, 68T01.

1. INTRODUCTION

Deep neural networks (DNNs) have made significant advancements in various fields, including computer vision, natural language processing, and game theory. Their remarkable capability to approximate complex functions has led to widespread application in tackling difficult problems in applied mathematics, particularly in solving partial differential equations (PDEs) and fractional PDEs.

Fractional PDEs are notable for their non-local behavior, making them effective models for processes that involve memory and hereditary effects. However, obtaining analytical solutions for fractional PDEs is often challenging due to the complexity of special functions, such as the Mittag-Leffler and Wright functions. Consequently, various numerical methods have been developed over the years to approximate these solutions such as finite difference, finite element, and spectral methods (see e.g. [1, 3, 23, 24, 27] and the references therein). Despite their usefulness, numerically solving fractional PDEs presents significant challenges. The high computational costs and memory requirements associated with the non-local behavior and singularities of fractional derivatives complicate the process. Traditional numerical methods, particularly for high-dimensional fractional PDEs, face substantial obstacles because they depend heavily on grid discretization, mesh generation, and the incorporation of physical laws, which increase computational costs and complexity.

Given these limitations, deep learning approaches, particularly physics-informed neural networks (PINNs), have emerged as promising alternatives [19]. PINNs offer a framework for solving PDEs by directly incorporating the physical laws described by these equations into the learning process through the loss function. The key idea is to adjust the loss function to embed these physical laws within the neural network, reconciling inconsistencies in data, such as initial

Received: 19 February 2025; Accepted: 26 January 2026.

* Corresponding author. Email: mokhtari@iut.ac.ir.

and boundary conditions or scattered measurements, with the constraints of the governing equations. Leveraging automatic differentiation (AD) capabilities in deep learning, PINNs can compute partial derivatives accurately, enabling them to solve PDEs and identify unknown governing parameters. Due to their powerful ability to embed physical laws directly within the neural network framework via automatic differentiation, PINNs have attracted considerable attention, spurring extensive research into their application for solving a wide array of PDEs across multiple scientific and engineering domains [5, 9, 26, 28, 29]. Despite the impressive capabilities of AD in accurately computing partial derivatives, certain challenges persist.

Automatic differentiation is not easily applicable to PDEs that involve non-local operators, such as fractional PDEs. To tackle this challenge, Pang et al. [17] introduced a new concept called fPINNs, which extends the framework of PINNs to solve space-time fractional advection-diffusion equations. The fPINNs approach is innovative as it combines automatic differentiation for integer-order operators with classical techniques, such as finite difference methods, for fractional-order operators. Recent advancements in fPINNs have highlighted their effectiveness in addressing fractional PDEs. In this context, an approximate formula for fractional derivatives, along with automatic differentiation, has been utilized to solve a specific class of fractional partial differential equations and their inverse problems. This approach offers significant advantages, including rapid convergence and robustness in addressing multidimensional challenges [4]. The TD-PINN has been introduced to model fractional water wave phenomena by discretizing the temporal domain using the BDF2 scheme [11]. Zhang et al. [30] proposed AWAO-fPINNs, an advanced approach that integrates a multi-output neural network architecture with adaptive loss weighting strategies to solve fractional partial integro-differential equations. Recently, RBF-fPINNs have been introduced [14], representing an innovative methodology that incorporates radial basis functions (RBFs) to enhance the accuracy of solutions to fractional mobile-immobile problems. This novel approach shows significant potential in addressing complex, high-dimensional fractional models, thus broadening the applicability and effectiveness of fPINNs across various scientific and engineering fields. Furthermore, dPINNs extend the capabilities of fPINNs to tackle distributed-order fractional problems, effectively addressing challenges in noisy and high-dimensional domains [20].

In this paper, we present an improved version of fPINNs specifically designed to tackle fractional reaction-diffusion problems, building on the foundational work in [14]. Our approach is complemented by an innovative neural network architecture designed to improve the performance of fPINNs in this context. The structure of this paper is organized as follows. In Section 2, we provide preliminary information, including an overview of the fractional reaction-diffusion problem. Section 3 highlights our key findings and introduces the innovative framework we developed. In Section 4, we detail our time discretization strategy. Section 5 presents numerical examples that demonstrate the effectiveness of our method. Finally, Section 6 concludes the paper with a summary of our main findings and discusses potential future research avenues.

2. PRELIMINARIES

Reaction-diffusion equations are essential tools in science and engineering, providing a versatile mathematical framework to model systems where substances interact through reactions and disperse spatially via diffusion [16]. In population biology, the reaction term models growth, while the diffusion term accounts for migration. This approach allows researchers to simulate and predict complex ecological dynamics.

These models are crucial for understanding a wide range of phenomena, including intricate patterns in chemical reactions, biological morphogenesis, and the spread of signals in neural networks. The fractional reaction-diffusion equation enhances this classical framework by incorporating fractional derivatives, which capture memory effects and non-local interactions [2]. This advancement enables a more accurate representation of systems where the standard assumption of local and instantaneous interactions is insufficient, thereby expanding the scope and applicability of reaction-diffusion models in addressing real-world challenges.

In light of this discussion, we will consider the following time fractional reaction-diffusion problem

$$\begin{aligned} D_t^\alpha u &= \Delta u - u + f(\mathbf{x}, t), & (\mathbf{x}, t) &\in \Omega \times (0, T], \\ u(\mathbf{x}, t) &= g(\mathbf{x}, t), & (\mathbf{x}, t) &\in \partial\Omega \times [0, T], \\ u(\mathbf{x}, t) &= u_0(\mathbf{x}), & \mathbf{x} &\in \Omega, \end{aligned} \tag{2.1}$$



on a bounded convex domain $\Omega \subset \mathbb{R}^2$. In (2.1), $\Delta = \partial_{x_1}^2 + \partial_{x_2}^2$ and D_t^α stands for the Caputo fractional derivative of order $\alpha \in (0, 1)$, which is defined as [8]

$$D_t^\alpha u(\cdot, t) = \frac{1}{\Gamma(1-\alpha)} \int_0^t (t-s)^{-\alpha} \frac{\partial}{\partial t} u(\cdot, s) ds. \tag{2.2}$$

As mentioned in [6], the solution u for problem (2.1) demonstrates weak singular behavior at initial time. In the upcoming sections, we will present the proposed framework to handle problem (2.1), incorporating the weak initial singularity property.

3. THE ADVANCED FPINN FRAMEWORK

To solve the fractional reaction-diffusion problem (2.1), we employ a radial basis function neural network, which uses radial basis functions as activation functions to approximate the solution. As an advanced three-layer feedforward neural network, the RBF model exhibits superior performance over back-propagation (BP) networks, delivering higher classification accuracy, faster learning, and improved approximation ability. Its intrinsic local approximation properties also enable a more rapid inversion process than those found in alternative neural network designs [10]. The theoretical foundation for employing RBF networks in function approximation is established by the universal approximation theorem for RBF networks. Specifically, Park and Sandberg proved that under mild conditions on the radial basis function, an RBF network with a single hidden layer can approximate any function in $L^p(\mathbb{R}^r)$ arbitrarily well [18].

Theorem 3.1. ([18, Theorem 1] *Universal Approximation in L^p spaces*)

Let $K : \mathbb{R}^r \rightarrow \mathbb{R}$ be an integrable bounded function such that K is continuous almost everywhere and

$$\int_{\mathbb{R}^r} K(x) dx \neq 0.$$

Then the family of RBF networks

$$S_K = \left\{ \sum_{i=1}^M w_i K\left(\frac{x - z_i}{\sigma}\right) : M \in \mathbb{N}, w_i \in \mathbb{R}, z_i \in \mathbb{R}^r, \sigma > 0 \right\},$$

is dense in $L^p(\mathbb{R}^r)$ for every $p \in [1, \infty)$.

In the proposed model, we specifically use Gaussian radial basis functions, defined as

$$\mathcal{R}_g(\|\mathbf{x} - \mathbf{c}\|) = e^{-\|\mathbf{x} - \mathbf{c}\|^2}.$$

With the Gaussian function serving as the activation function, the relationship between the input and output in the RBF neural network can mathematically be formulated as follows

$$\tilde{u}(\mathbf{x}) = \sum_{i=1}^m w_i e^{-\|\mathbf{x} - \mathbf{c}_i\|^2},$$

Here, m indicates the total number of radial basis functions (or hidden nodes), and $\|\cdot\|$ signifies the Euclidean norm. The term w_i represents the connection weight from the i -th hidden unit to the output unit, while \mathbf{c}_i denotes the center associated with the i -th node. In Figure 1, the block diagram of the RBF neural network is illustrated, showing its architecture, including the input layer, the hidden layer with Gaussian radial basis functions, and the output layer.

We define the notation $\lambda = \{w_i, \mathbf{c}_i\}_{i=1}^m$ to include all trainable parameters of the network. As a result, the output of the network can be rewritten as $u_{Net}(\mathbf{x}, t; \lambda)$. In this framework, we can choose a form of the approximate solution that inherently satisfies both the initial and boundary conditions. Specifically, we define the approximate solution as

$$\tilde{u}(\mathbf{x}, t) = t \rho(\mathbf{x}) u_{Net}(\mathbf{x}, t; \lambda) + g(\mathbf{x}, t),$$

where $\rho(\mathbf{x})$ is an auxiliary function preselected to ensure that the boundary and initial conditions are automatically satisfied. A well-chosen $\rho(\mathbf{x})$ ensures that the solution inherently satisfies these constraints. Specifically, $\rho(\mathbf{x})$ must satisfy the following conditions [13]

$$\rho(\mathbf{x}) = 0, \quad \mathbf{x} \in \partial\Omega, \quad \text{and} \quad \rho(\mathbf{x}) > 0, \quad \mathbf{x} \in \Omega - \partial\Omega.$$



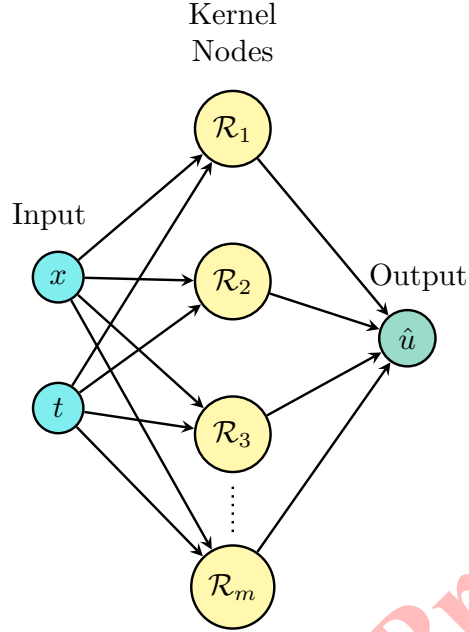


FIGURE 1. Architecture of the RBF neural networks.

Typically, if the boundary $\partial\Omega$ has a simple geometric structure, an analytical formulation of $\rho(\mathbf{x})$ can be determined [7]. However, for more complex domains, such as L-shaped, spherical, or star-shaped regions, it is difficult to obtain an analytical formula of $\rho(\mathbf{x})$ and, a different approach is needed to accurately model $\rho(\mathbf{x})$. In [14], we thoroughly discussed and examined in detail such cases and considered an alternative methodology to handle the complexities of these geometries.

In this scenario, we specify the operator $\mathcal{L}[u(\mathbf{x}, t)]$ in the following manner, where \mathcal{L} describes the dynamics of the underlying fractional reaction-subdiffusion problem

$$\mathcal{L}[u(\mathbf{x}, t)] := D_t^\alpha u(\mathbf{x}, t) - \Delta u(\mathbf{x}, t) + u(\mathbf{x}, t).$$

We can classify the operators that comprise \mathcal{L} into two distinct categories:

$$\mathcal{L} := \mathcal{L}_{\text{nonAD}} + \mathcal{L}_{\text{AD}} := D_t^\alpha + \Delta - u.$$

The \mathcal{L}_{AD} component contains operators that can be evaluated using automatic differentiation. Conversely, $\mathcal{L}_{\text{nonAD}}$ consists of operators that cannot be computed through automatic differentiation. For this term, numerical approximation techniques are required, which will be discussed in greater detail later.

To effectively train the model, we delineate the loss function with the following formalism

$$L(\lambda) = \frac{1}{|\psi|} \sum_{(\mathbf{x}, t) \in \psi} \left(\mathcal{L}_{\text{nonAD}}[\tilde{u}(\mathbf{x}, t)] + \mathcal{L}_{\text{AD}}[\tilde{u}(\mathbf{x}, t)] - f(\mathbf{x}, t) \right)^2, \quad (3.1)$$

wherein, $|\psi|$ the number of training points in the training sets $\psi \subset \Omega \times (0, T]$. The training process for the proposed model in this problem involves minimizing the loss function (3.1) concerning λ using the L-BFGS optimizer [12] to find the optimal parameters for the RBF network.

4. TIME DISCRETIZATION

In this section, we concentrate on discretizing the Caputo fractional derivative (2.2) using the L1 formula [15]. While S-type formulas [21, 22] offer potential accuracy improvements, we opt for the L1 formula here, deferring the



examination of S-type formulas for future research. For a positive N , let $\tau = \frac{T}{N}$, and we $t_k = k\tau$, $0 \leq k \leq N$. The L1 formula yields the following result:

$$\begin{aligned} D_t^\alpha \tilde{u}(\cdot, t_n) &= \frac{1}{\Gamma(1-\alpha)} \sum_{k=0}^{n-1} \int_{t_k}^{t_{k+1}} (t_n - s)^{-\alpha} \frac{\partial \tilde{u}}{\partial t}(\cdot, s) \, ds \\ &\simeq \frac{1}{\Gamma(1-\alpha)} \sum_{k=0}^{n-1} \frac{\tilde{u}(\cdot, t_{k+1}) - \tilde{u}(\cdot, t_k)}{\tau} \times \int_{t_k}^{t_{k+1}} (t_n - s)^{-\alpha} \, ds \\ &\simeq a_0^{(\alpha)} \tilde{u}(\cdot, t_n) - \sum_{k=1}^{n-1} (a_{n-k-1}^{(\alpha)} - a_{n-k}^{(\alpha)}) \tilde{u}(\cdot, t_k) - a_{n-1}^{(\alpha)} \tilde{u}(\cdot, t_0), \end{aligned}$$

therein $a_l^{(\alpha)} = \frac{\tau^{-\alpha}}{\Gamma(2-\alpha)} ((l+1)^{1-\alpha} - l^{1-\alpha})$ for $l \geq 0$.

5. NUMERICAL RESULTS

This section aims to demonstrate the proposed approach’s capability in solving fractional PDEs by providing two examples: one on a rectangular domain and another on a circular domain. These examples will illustrate the method’s effectiveness in handling different geometric configurations and highlight its versatility in solving fractional PDEs. The accuracy of the developed model is evaluated by calculating the relative L_2 error between the predicted solution and the exact solution

$$\text{Relative } L_2 \text{ error} = \frac{\|\tilde{u}(\mathbf{x}_{test}, t_{test}) - u(\mathbf{x}_{test}, t_{test})\|_2}{\|u(\mathbf{x}_{test}, t_{test})\|_2},$$

where u_{NN} and u correspond to the predicted and exact solutions, respectively. In this study, we adopt the Gaussian function as an activation function within the advanced fPINN framework. This choice is intended to enhance the model’s capability to accurately capture the dynamics of solutions for fractional PDEs. The simulation was executed on a single NVIDIA Tesla V100 GPU, capitalizing on the computational efficiency of the PyTorch framework.

Example 5.1. We examine the fractional reaction-diffusion Equation (2.1) defined on a bounded two-dimensional rectangular domain Ω , incorporating the following source term

$$f(\mathbf{x}, t) = (3t^\alpha + \Gamma(1 + \alpha)) \sin(x_1) \sin(x_2).$$

The analytical solution to this problem is provided by

$$u(\mathbf{x}, t) = t^\alpha \sin(x_1) \sin(x_2).$$

Next, we present the auxiliary function $\rho(\mathbf{x})$ defined as

$$\rho(\mathbf{x}) = x_1(\pi - x_1)x_2(\pi - x_2),$$

where $\mathbf{x} = (x_1, x_2)$ represents the coordinates in the domain Ω . In the numerical experiments, we sample 50 training points on the boundary and 150 points within the interior of the spatial domain using Latin hypercube sampling (LHS) to ensure a well-distributed set of points. LHS is a statistical method introduced by Michael Stein [25] for efficiently constructing a representative sample from a multi-dimensional probability distribution. The core idea behind LHS is to stratify the probability distribution of each variable, ensuring that the sample points are distributed more evenly across the input space compared to simple random sampling methods. In the LHS approach, the search space is first divided into equal intervals along each dimension. Then, for each dimension, a random selection is made from within the intervals, ensuring that the entire range of each dimension is sampled in a way that reflects its distribution. These selected intervals are then paired randomly across the different dimensions to form a set of sampling regions. Afterward, a sample is randomly generated within these regions, ensuring that all parts of the input space are explored systematically. The primary advantage of LHS is its ability to generate a sample that better represents the full range of the input space with fewer points than simple random sampling. This makes it an efficient method for high-dimensional and computationally expensive problems. For spatial testing, the computational domain is partitioned



using 56 uniformly distributed nodes along the boundary and 169 within the interior. In Figure 2, the distribution of training and test points is presented within the rectangular domain. The training results, summarized in Table 1,

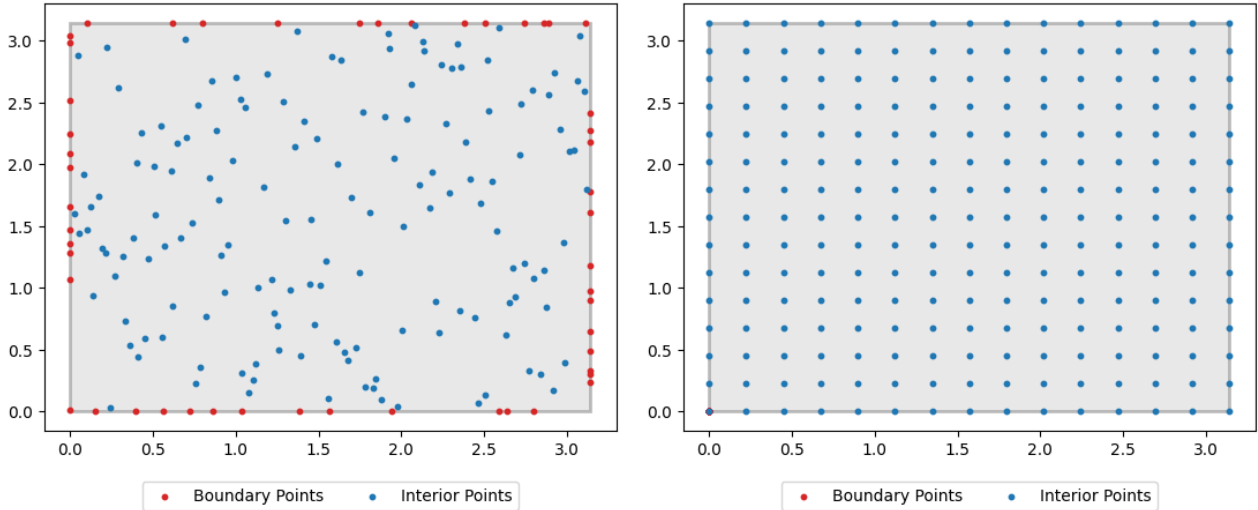


FIGURE 2. Data points for training (left) and testing (right).

show the relative L_2 errors for training and testing datasets across various fractional orders α and different temporal training nodes N . The results reveal a consistent decline in relative L_2 error with increasing N , reflecting enhanced

TABLE 1. The relative L_2 errors of advanced fPINN for different α .

α	N	Train: L_2 error	Test: L_2 error
0.9	10	8.5691e-03	9.4007e-03
	20	5.0781e-03	5.2961e-03
	40	2.9811e-03	3.0333e-03
	80	1.8270e-03	1.8407e-03
0.5	10	2.0597e-02	3.1386e-02
	20	1.3174e-02	1.7935e-02
	40	8.3898e-03	1.0132e-02
	80	7.0808e-03	7.3436e-03
0.1	10	8.8966e-03	9.2007e-02
	20	1.8379e-02	6.5814e-02
	40	3.2803e-02	5.3172e-02
	80	4.7975e-02	5.2555e-02

accuracy for training and testing scenarios. Generally, higher fractional orders α correspond to significantly lower error rates, particularly at larger N values. In contrast, for lower fractional orders, while the errors remain relatively stable, a noticeable increase is evident at higher N . These findings underscore the effectiveness of the Gaussian activation function in achieving convergence and accuracy within the advanced fPINN framework. Figures 3 through 5 illustrate the performance of the proposed advanced fractional Physics-Informed Neural Network (fPINN) framework for different values of the fractional order parameter α . In Figure 3, for $\alpha = 0.9$, the expected solution aligns with the exact solution showcasing the model's high accuracy for larger fractional orders. In the case of $\alpha = 0.5$ in Figure 4, the framework provides an excellent approximation even in intermediate fractional scenarios. In Figure 5, for $\alpha = 0.1$,



despite the increased complexity of lower fractional orders, the model effectively predicts the solution and the overall prediction remains consistent with the exact solution. These results highlight the ability of the proposed framework to handle varying levels of complexity in fractional reaction-diffusion problems with reliable accuracy across a range of fractional orders.

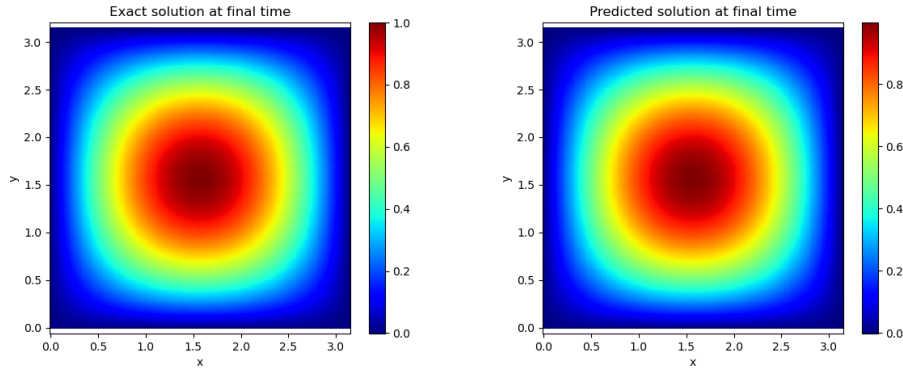


FIGURE 3. Exact solution, and predicted solution at final time for $N = 10$ and $\alpha = 0.9$.

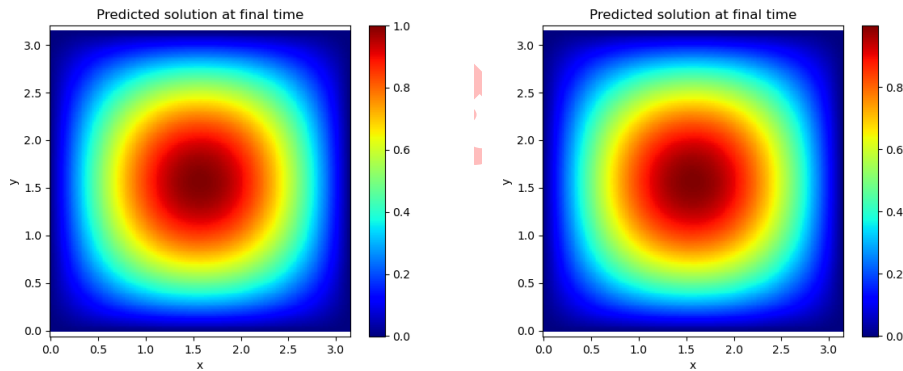


FIGURE 4. Exact solution, and predicted solution at final time for $N = 10$ and $\alpha = 0.5$.

Example 5.2. In order to further demonstrate the performance and versatility of the proposed approach, we next focus on the circle as the domain of interest. The circular geometry introduces distinct challenges compared to simpler, rectangular domains, as it involves handling boundary conditions and solutions on a non-rectilinear domain. This non-linear structure requires special consideration for both the source term and the auxiliary function to ensure accurate predictions.

We investigate the fractional reaction-diffusion Equation (2.1) on this circular domain. The source-term f and exact solution are given by

$$\begin{aligned}
 f(\mathbf{x}, t) &= t^a \left((-3x_1^2 - 3x_2^2 + 7) \sin(x_1 + x_2) + 4(x_1 + x_2) \cos(x_1 + x_2) \right) \\
 &\quad - \Gamma(a + 1) (x_1^2 + x_2^2 - 1) \sin(x_1 + x_2), \\
 u(\mathbf{x}, t) &= t^a (-x_1^2 - x_2^2 + 1) \sin(x_1 + x_2).
 \end{aligned}$$



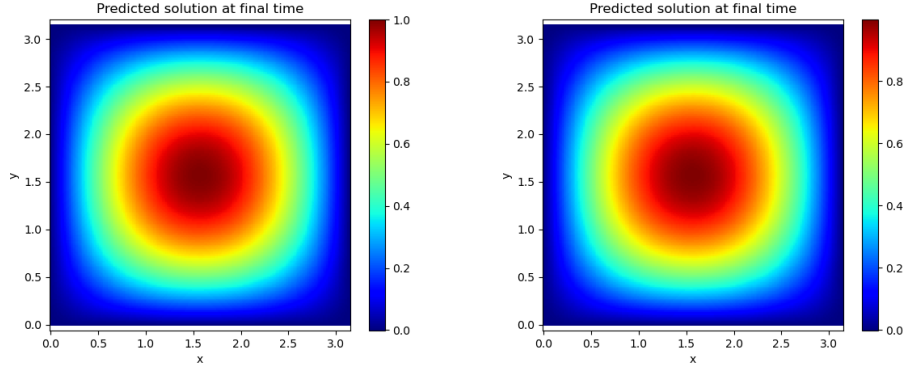


FIGURE 5. Exact solution, and predicted solution at final time for $N = 10$ and $\alpha = 0.1$.

For this case, we introduce the auxiliary function $\rho(\mathbf{x})$, defined as

$$\rho(\mathbf{x}) = (1 - \|\mathbf{x}\|_2^2).$$

This section concludes with the evaluation of the results from Example 5.2, where the effectiveness of the proposed model in solving fractional reaction-diffusion equations on a circular domain is assessed. Table 2 presents the relative L_2 errors for different values of α and varying numbers of points N , computed for both training and testing data. As clearly demonstrated, the proposed model exhibits excellent performance, with the L_2 errors decreasing significantly as N increases. This improvement highlights the convergence and robustness of the method. For larger values of α , the model achieves particularly high accuracy, while for smaller values of α , the results still show a consistent and notable improvement as N grows.

These findings underscore the capability of the proposed model to handle complex geometries, such as circular domains, effectively. The results confirm that the method is accurate and reliable, even for challenging cases with non-rectangular domains. Figure 6 presents the relative L_2 errors on a circular domain at the final time for different

TABLE 2. (Example 5.2) The relative L_2 errors utilizing different temporal training.

α	N	Train: L_2 error	Test: L_2 error
0.9	10	2.4943e-03	3.1466e-03
	20	1.6281e-03	1.8145e-03
	40	1.2146e-03	1.2674e-03
0.5	10	6.1406e-03	1.8806e-02
	20	4.7454e-03	1.0596e-02
	40	5.8471e-03	8.0607e-03
0.1	10	2.9131e-03	8.7977e-02
	20	1.2730e-02	5.9106e-02
	40	3.1729e-02	5.1449e-02

values of the fractional order parameter α and varying temporal training steps, denoted by N . The figure illustrates how the model performs for three distinct fractional orders: $\alpha = 0.1$, $\alpha = 0.5$, and $\alpha = 0.9$. As the number of temporal training steps N increases, there is a clear reduction in the relative L_2 error, demonstrating that the model becomes more accurate with higher temporal resolution. Notably, the model performs exceptionally well for higher fractional orders $\alpha = 0.9$, yielding smaller errors even with fewer temporal steps. On the other hand, for smaller fractional orders such as $\alpha = 0.1$, the model still shows a steady improvement in accuracy as N increases, underscoring the model's robustness across a range of fractional order values. These results highlight the ability of the proposed model



to effectively handle complex dynamics on a circular domain, with significant improvements in accuracy as temporal resolution increases.

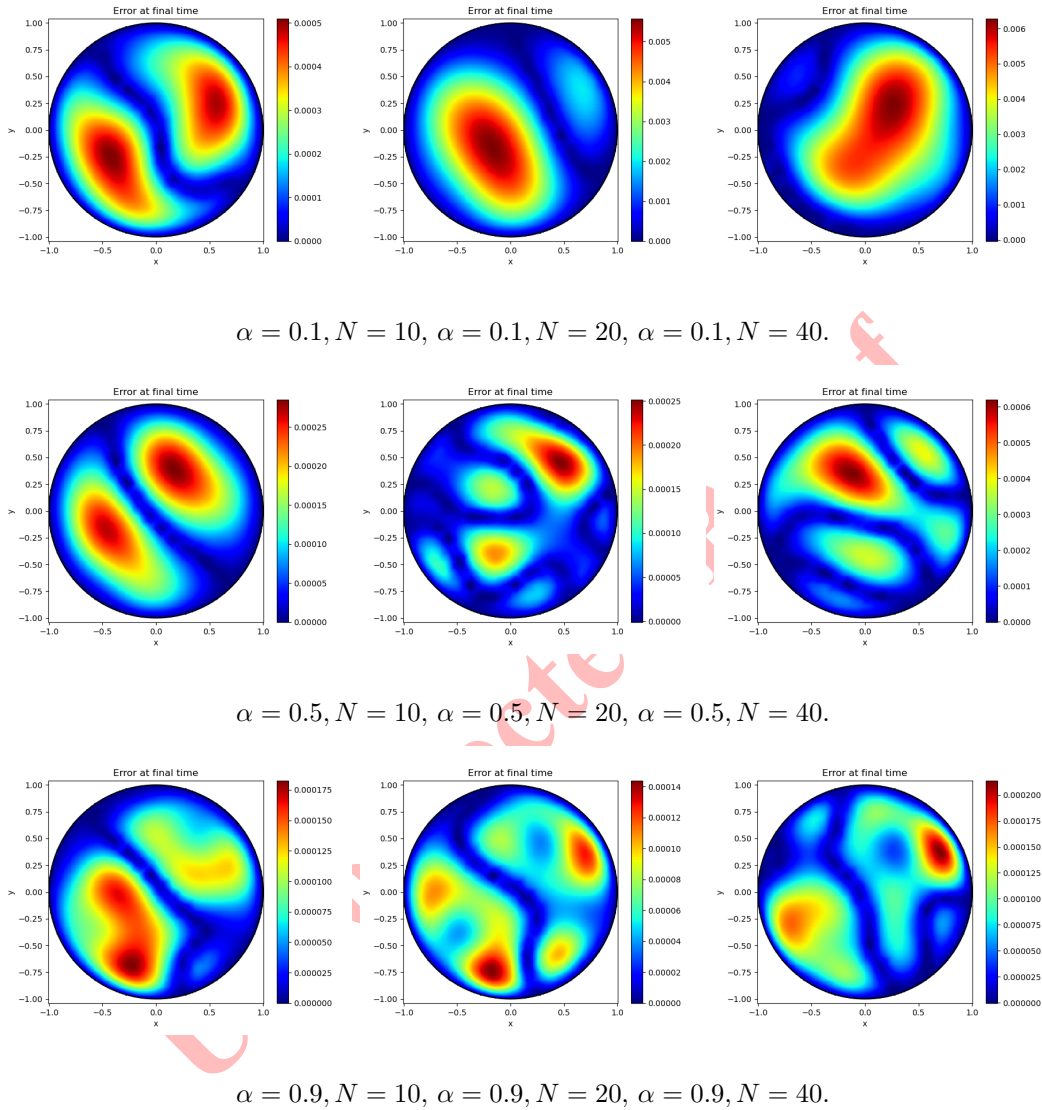


FIGURE 6. (Example 5.2) Relative L_2 errors on the circular domain.

6. CONCLUSION

We have introduced a novel fPINN framework for effectively addressing fractional reaction-diffusion problems. The proposed framework has demonstrated substantial improvements in the context of two-dimensional domains, specifically in rectangular and circular domains. Moreover, by introducing appropriate auxiliary functions based on the geometry of the domain, the model can naturally satisfy the required initial and boundary conditions. This framework also shows great potential in handling solutions with weak singularities, which are often encountered in



fractional models. Looking ahead, we aim to extend this approach to solve wide classes of nonlinear problems involving significant nonlinearity, which the classical fPINN approach often struggles with.

CONFLICT OF INTEREST STATEMENT

The authors declare no potential conflict of interests.

REFERENCES

- [1] A. A. Alikhanov, M. S. Asl, and C. Huang, *Stability analysis of a second-order difference scheme for the time-fractional mixed sub-diffusion and diffusion-wave equation*, *Fractional Calculus and Applied Analysis*, 27(1) (2024), 102-123.
- [2] B. Baeumer, M. Kovács, and M. M. Meerschaert, *Numerical solutions for fractional reaction–diffusion equations*, *Computers & Mathematics with Applications*, 55(10) (2008), 2212-2226.
- [3] L. Cai, J. Cao, F. Jing, and Y. Wang, *A fast time integral finite difference method for a space-time fractional FitzHugh-Nagumo monodomain model in irregular domains*, *Journal of Computational Physics*, 501 (2024), 112744.
- [4] X. Fang, L. Qiao, F. Zhang, and F. Sun, *Explore deep network for a class of fractional partial differential equations*, *Chaos, Solitons & Fractals*, 172 (2023), 113528.
- [5] B. Gao, R. Yao, and Y. Li, *Physics-informed neural networks with adaptive loss weighting algorithm for solving partial differential equations*, *Computers & Mathematics with Applications*, 181 (2025), 216-227.
- [6] T. Kumari and P. Roul, *An efficient computational technique for solving a time-fractional reaction-subdiffusion model in 2D space*, *Computers & Mathematics with Applications*, 160 (2024), 191-208.
- [7] P. L. Lagari, L. H. Tsoukalas, S. Safarkhani, and I. E. Lagaris, *Systematic construction of neural forms for solving partial differential equations inside rectangular domains, subject to initial, boundary and interface conditions*, *International Journal on Artificial Intelligence Tools*, 29 (05)(2020), 2050009.
- [8] C. Li and F. Zeng, *Numerical methods for fractional calculus*, CRC Pres, (2015).
- [9] Y. Liang, R. Niu, J. Yue, and M. Lei, *A physics-informed recurrent neural network for solving time-dependent partial differential equations*, *International Journal of Computational Methods*, 21(10) (2024), 2341003.
- [10] W. M. Lin, C. D. Yang, J. H. Lin, and M. T. Tsay, *A fault classification method by RBF neural network with OLS learning procedure*, *IEEE Transactions on Power Delivery*, 16(4) (2001), 473-477.
- [11] W. Liu, Y. Liu, and H. Li, *Time difference physics-informed neural network for fractional water wave models*, *Results in Applied Mathematics*, 17 (2023), 100347.
- [12] D. C. Liu and J. Nocedal, *On the limited memory BFGS method for large scale optimization*, *Mathematical programming*, 45(1) (1989), 503-528.
- [13] L. Lu, R. Pestourie, W. Yao, Z. Wang, F. Verdugo, and S. G. Johnson, *Physics-informed neural networks with hard constraints for inverse design*, *SIAM Journal on Scientific Computing*, 43(6) (2021), B1105-B1132.
- [14] M. Mohammadi, R. Mokhtari, and M. Ramezani, *RBF-fPINNs: Radial basis function-enhanced fractional physics-informed neural networks*, To appear in *Engineering with Computers*.
- [15] K. Oldham and J. Spanier, *The Fractional Calculus*, New York: Academic Press, (1974).
- [16] K. M. Owolabi, S. Jain, E. Pindza, and E. Mare, *Comprehensive numerical analysis of time-fractional reaction-diffusion models with applications to chemical and biological phenomena*, *Mathematics*, 12(20) (2024), 3251.
- [17] G. Pang, L. Lu, and G. E. Karniadakis, *fPINNs: Fractional physics-informed neural networks*, *SIAM Journal on Scientific Computing*, 41(4) (2019), A2603-A2626.
- [18] J. Park and I. W. Sandberg, *Universal approximation using radial-basis-function networks*, *Neural computation*, 3(2) (1991), 246-257.
- [19] M. Raissi, P. Perdikaris, and G. E. Karniadakis, *Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations*, *Journal of Computational physics*, 378 (2019), 686-707.



- [20] M. Ramezani, M. Mohammadi, and R. Mokhtari, *dPINNs: A physics-informed framework for forward and inverse problems governed by distributed-order derivatives*, Engineering Analysis with Boundary Elements, 179 (2025), 106418.
- [21] M. Ramezani and R. Mokhtari, *A novel high-order finite-difference method for the time-fractional diffusion equation with smooth/nonsmooth solutions*, Bulletin of the Iranian Mathematical Society, 48(6) (2022), 3987-4013.
- [22] M. Ramezani, R. Mokhtari, and G. Haase, *Some high order formulae for approximating Caputo fractional derivatives*, Applied Numerical Mathematics, 153 (2020), 300-318.
- [23] M. Ramezani, R. Mokhtari, and Y. Yan, *Correction of a high-order numerical method for approximating time-fractional wave equation*, Journal of Scientific Computing, 100(3) (2024), 71.
- [24] A. Seal, S. Natesan, and S. Toprakseven, *A dimensional-splitting weak Galerkin finite element method for 2D time-fractional diffusion equation*, Journal of Scientific Computing, 98(3) (2024), 56.
- [25] M. Stein, *Large sample properties of simulations using latin hypercube sampling*, Technometrics, 29(32) (1987), 143-151.
- [26] W. Wu, S. Duan, Y. Sun, Y. Yu, D. Liu and D. Peng, *Deep fuzzy physics-informed neural networks for forward and inverse PDE problems*, Neural Networks, 181 (2025), 106750.
- [27] Y. Yang, J. Huang, and H. Li, *An α -robust analysis of finite element method for space-time fractional diffusion equation*, Numerical Algorithms, (2024), 1-26.
- [28] B. Yuan, H. Wang, A. Heitor, and X. Chen, *f-PICNN: A physics-informed convolutional neural network for partial differential equations with space-time domain*, Journal of Computational Physics, 515 (2024), 113284.
- [29] H. Zhang, L. Jiang, X. Chu, Y. Wen, L. Li, J. Liu, Y. Xiao, and L. Wang, *Combining physics-informed graph neural network and finite difference for solving forward and inverse spatiotemporal PDEs*, Computer Physics Communications, 308 (2025), 109462.
- [30] J. Zhang, Y. Zhao, and Y. Tang, *Adaptive loss weighting auxiliary output fPINNs for solving fractional partial integro-differential equations*, Physica D: Nonlinear Phenomena, 460 (2024), 134066.

Uncorrected Proof

