Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

A Bernstein Polynomial-Based Neural Network Framework for Solving Fractional Tumor Growth Mathematical Model

Paras Varshney
Kedarnath Girdharilal Khatri College
Govind Raj Naunyal
Kedarnath Girdharilal Khatri College
Sachin Kumar
sachinkr.rs.mat17@itbhu.ac.in

Government Degree College

Research Article

Keywords:

Posted Date: July 28th, 2025

DOI: https://doi.org/10.21203/rs.3.rs-7077249/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Version of Record: A version of this preprint was published at Zeitschrift für angewandte Mathematik und Physik on March 2nd, 2026. See the published version at https://doi.org/10.1007/s00033-026-02743-7.

A Bernstein Polynomial-Based Neural Network Framework for Solving Fractional Tumor Growth Mathematical Model

Paras Varshney[1], Govind Raj Naunyal[1], and Sachin Kumar*[2]

[1]Department of Mathematics, Kedarnath Girdharilal Khatri College, Moradabad 202412, India.
[2]Department of Mathematics, Government Degree College, Budaun 243601, India.

Abstract

This paper presents a novel approach for solving fractional-order differential equations using Bernstein Neural Networks (BNNs). The proposed architecture comprises input, hidden, and output layers, where Bernstein polynomials of varying degrees are employed as basis functions. Activation functions are incorporated in the hidden layers to enhance nonlinearity, while the network's output is trained to approximate the solution of the differential equation. The unknown solution is modeled as a function of trainable parameters (weights), which are optimized via the backpropagation algorithm. To evaluate the effectiveness and accuracy of the method, benchmark problems with known exact solutions are considered. Comparative analysis with existing numerical techniques demonstrates that the BNN-based approach yields superior accuracy and computational performance, confirming its potential for efficiently solving complex fractional-order systems.

1 Introduction

Fractional calculus, the study of derivatives and integrals of arbitrary (non-integer) order, has emerged as a powerful mathematical tool for modeling complex systems exhibiting memory and hereditary properties. Unlike classical integer-order models, fractional-order models capture long-range temporal correlations and nonlocal behaviors observed in many physical and biological processes. Applications of fractional calculus span across diverse domains, including viscoelastic materials, signal processing, control systems, and anomalous diffusion. In particular, fractional operators such as the Caputo and Riemann–Liouville derivatives provide the flexibility needed to describe real-world systems more accurately, especially those governed by sub-diffusive or super-diffusive transport mechanisms [1–4].

*Corresponding author: sachinraghav522@gmail.com

The study of fractional differential equations (FDEs) has grown in prominence due to their ability to incorporate such memory effects into mathematical models. However, solving FDEs analytically remains challenging except for a limited class of problems with simple structures. Consequently, numerous numerical methods have been developed for approximating solutions to FDEs. These include finite difference methods, spectral techniques, the Adomian decomposition method, and operational matrix approaches [5–7]. Recent advancements [8] also focus on the development of fast algorithms such as the Fast L1 and L1-2 schemes to efficiently compute Caputo-type derivatives. Despite these advancements, high-dimensional or nonlinear systems still pose computational challenges, motivating the exploration of data-driven and neural network-based methods for solving FDEs.

Tumor cell modeling plays a crucial role in understanding the progression and treatment of cancer [9]. Mathematical models can help simulate tumor dynamics under various conditions, including the presence or absence of therapeutic agents. Fractional-order models have been increasingly employed to represent tumor growth and interaction with the immune system, as they can better capture the heterogeneity, delayed response, and memory effects inherent in biological tissues. Incorporating fractional dynamics in tumor models allows for more accurate simulations of cancer cell proliferation, invasion, and response to chemotherapeutic treatments, leading to more reliable predictions and treatment optimization strategies [10–13].

In recent years, neural networks have gained significant attention as flexible, mesh-free, and highly accurate tools for solving differential equations. Physics-Informed Neural Networks (PINNs) and other variants have been successfully applied to fractional-order systems due to their ability to approximate unknown functions while incorporating differential operators into the training loss function. In this study, we propose the use of *Bernstein Neural Networks* (BNNs), which utilize Bernstein polynomial basis functions in the hidden layer, offering smooth approximation properties and improved numerical stability [14]. The network is trained via backpropagation to minimize the residual of the FDEs. Previous works have demonstrated the effectiveness of BNNs in various engineering and biomedical contexts [15, 16], motivating their application here for solving a fractional-order tumor model with and without chemotherapy.

In this study, the authors present a new method using a Bernstein Neural Network (BNN), where three hidden layer uses Bernstein polynomials of different degrees and hyperbolic of tangent as an activation function. The network is made up of twelve input layer, three hidden layer and four output layer. Each neuron (or perceptron) in the hidden layer applies a Bernstein polynomial with adjustable weight, and the output layer combines these using trainable weights. These weights are initially chosen at random and then adjusted during training using a backpropagation algorithm, which will be explained in later sections. The main idea of using a BNN is to approximate the solution of a system of fractional differential equations (FDEs). This is done by treating the BNN output as the approximate solution. A residual function is then created using the given FDEs along with initial or boundary conditions. The goal is to train the network so that this residual becomes as small as possible, effectively solving the FDE system. The proposed method is tested on a system of differential equations and is shown to perform better than existing methods through detailed comparisons and analysis.

2 Preliminaries

This section presents the essential mathematical foundations and notational conventions relevant to fractional calculus and its application in the neural network-based modeling of fractional partial differential equations (FPDEs). Particular emphasis is placed on the Caputo time-fractional derivative, which is widely utilized in modeling systems with memory and hereditary characteristics—common in various physical and biological processes.In the present study, we adopt the Caputo fractional derivative framework for capturing the temporal dynamics inherent to the modeled system.

2.1 Riemann - Liouville fractional derivative [17]

The Riemann–Liouville fractional derivative generalizes the classical derivative to non-integer orders. For $\alpha > 0$, the Riemann–Liouville fractional derivative of order α of a function $f(x)$ (with lower limit $\mathfrak{a}$) is defined as

$$D^{\alpha}_{\mathfrak{a}} f(x) = \frac{1}{\Gamma(n-\alpha)} \frac{d^n}{dx^n} \int_{\mathfrak{a}}^{x} (x-\xi)^{n-\alpha-1} f(\xi)\, d\xi$$

where$\Gamma(\cdot)$ is the Gamma function and $n = \lceil \alpha \rceil$ (the smallest integer $\geq \alpha$). This definition requires f to be sufficiently smooth on $[\mathfrak{a}, x]$ so that the integral and derivative exist.

2.2 Caputo Fractional Derivative [18]

The Caputo fractional derivative is among the most widely employed formulations in the modeling of physical phenomena, primarily due to its compatibility with classical initial conditions. Unlike other definitions, it enables the incorporation of initial values expressed in terms of integer-order derivatives, making it particularly advantageous for physical and engineering applications.

Let $f(x, s)$ be a sufficiently smooth function. The Caputo fractional derivative of order $\beta \in (0, 1)$ with respect to the variable s is defined as:

$${}^{C}_{a}D^{\beta}_{s} f(x, s) = \frac{1}{\Gamma(1-\beta)} \int_{a}^{s} \frac{\partial f(x, \xi)}{\partial \xi} \frac{d\xi}{(s-\xi)^{\beta}},$$

where $\Gamma(\cdot)$ denotes the Gamma function. This definition is particularly suitable for physical systems with memory and hereditary properties, as it expresses the fractional derivative as a convolution of the first-order derivative with a power-law kernel.

The Caputo fractional derivative of order $\mathfrak{a}$ of the polynomial p^q is given by:

$${}^{C}_{0}D^{\mathfrak{a}}_{p} p^q = \begin{cases} 0, & \text{if } q \in \{0, 1, 2, \ldots, \lceil \mathfrak{a} \rceil - 1\}, \\ \dfrac{\Gamma(q+1)}{\Gamma(q+1-\mathfrak{a})} p^{q-\mathfrak{a}}, & \text{if } q \in \mathbb{N},\ q \geq \lceil \mathfrak{a} \rceil, \end{cases} \tag{1}$$

where $\lceil \mathfrak{a} \rceil = \min\{n \in \mathbb{Z} \mid n \geq \mathfrak{a}\}$ denotes the ceiling of $\mathfrak{a}$, and $\mathbb{Z}$ is the set of integers.

2.3 Characteristic of Bernstein Polynomial

The Bernstein basis polynomials of degree N are defined as:

$$B_{k,N}(\xi) = \binom{N}{k} \xi^k (1-\xi)^{N-k}, \quad \text{for } k = 0, 1, \ldots, N, \tag{2}$$

where $\binom{N}{k} = \dfrac{N!}{k!(N-k)!}$ denotes the binomial coefficient.

A general Bernstein polynomial of degree N can be expressed as:

$$\mathcal{B}_N(\xi) = \sum_{k=0}^{N} \beta_k \, B_{k,N}(\xi), \tag{3}$$

where β_k are the Bernstein coefficients, and $\xi \in [0, 1]$.

Equation (3) can equivalently be written using the factorial form of the binomial coefficient as:

$$B_{k,N}(\xi) = \frac{N!}{k!(N-k)!} \, \xi^k (1-\xi)^{N-k}, \quad \text{for } k = 0, 1, \ldots, N. \tag{4}$$

The Caputo fractional derivative of order $\alpha \in (0, 1)$ of the Bernstein basis polynomial $B_{k,N}(\xi)$ with respect to ξ is given by:

$${}_0^C D_\xi^\alpha B_{k,N}(\xi) = \frac{N!}{k!(N-k)!} \cdot \begin{cases} 0, & \text{if } k \in \{0, 1, \ldots, \lceil \alpha \rceil - 1\}, \\ \dfrac{\Gamma(k+1)}{\Gamma(k+1-\alpha)} \, \xi^{k-\alpha} (1-\xi)^{N-k}, & \text{if } k \in \mathbb{N},\ k \geq \lceil \alpha \rceil, \end{cases} \tag{5}$$

where $\lceil \alpha \rceil$ denotes the smallest integer greater than or equal to α, and $\Gamma(\cdot)$ is the Gamma function.

3 Description of the Model Representing the Chemotherapy Effect on Tumor Cell Dynamics

In this section, we consider a mathematical model describing the interaction dynamics of tumor cells under the influence of chemotherapy. The system consists of four coupled time-fractional partial differential equations (FPDEs), formulated using the Caputo fractional derivative of order $\alpha \in (0, 1)$, and is equipped with suitable initial and boundary conditions.

The model incorporates the concentration of three types of cells: normal cells (N), tumor cells (T), and immune cells (I). These variables evolve over time and space through both reaction and diffusion processes. The equations collectively capture the behavior of tumor growth and its interaction with normal and immune cells, as well as the effects of chemotherapy. The focus of this study is on analyzing the time-fractional version of the model, represented by equation (6) − (9), which introduces memory effects into the dynamics—an essential feature when modeling biological systems.

The general form of our tumor model is [19]:

$$D_t^\alpha N(x,t) = D_N N_{xx} - a_3(1-e^{-U})N - c_4 NT - \omega_1 N^2 + \lambda_1 N + f_N(x,t) \tag{6}$$
$$D_t^\alpha T(x,t) = D_T T_{xx} - a_2(1-e^{-U})T - c_3 NT - c_2 IT - \omega_2 T^2 + \lambda_2 T + f_T(x,t) \tag{7}$$
$$D_t^\alpha I(x,t) = D_I I_{xx} + a_1 e^{-U} I - c_1 IT - \lambda_1 I + \frac{\rho IT}{\mu + T} + \epsilon + f_I(x,t) \tag{8}$$
$$D_t^\alpha U(x,t) = D_U U_{xx} - d_2 U + \varphi(t) + f_U(x,t) \tag{9}$$

with initial and boundary condition as follows and D_t^α is the Caputo derivative and $f_k(x,t)$ are known forcing terms which are derived from the following exact solution

$$N(x,t) = 0.2\, e^{-t} e^{-2x^2}, \quad T(x,t) = 1 - 0.75\, e^{-t}\mathrm{sech}(x),$$
$$I(x,t) = e^{-t}\mathrm{sech}(x), \quad U(x,t) = -0.235\, e^{-t}\mathrm{sech}^2(x) + 0.375.$$

and initial condition are

$$\begin{aligned} T(x,0) &= h_2(x) = 1 - 0.75\,\mathrm{sech}\, x, \\ N(x,0) &= h_1(x) = 0.2e^{-2x^2}, \\ U(x,0) &= h_4(x) = \mathrm{sech}\, x. \\ I(x,0) &= h_3(x) = 0.375 - 0.235\,\mathrm{sech}^2\, x, \end{aligned} \tag{10}$$

The model includes several biologically relevant parameters, each with specific physical interpretations. A detailed description of these parameters is provided in Table (1)

Table 1: Description and numerical values of model parameters.

Parameter	**Description**	**Value**
a_1, a_2, a_3	Cell death induced by drug	0.2, 0.3, 0.1
c_1, c_2, c_3, c_4	Competition coefficients	1, 0.55, 0.9, 1
d_2	Drug degradation rate	1.0
ω_1, ω_2	Logistic growth limiting terms	0.81, 1.1
λ_1, λ_2	Proliferation/growth rates	0.9, 0.8
ρ	Immune response rate	0.2
μ	Saturation threshold	0.3
ϵ	Immune influx term	0.33
$\varphi(t)$	Chemotherapeutic drug	0
D_N, D_T, D_I, D_U	Diffusion coefficients	0.001 each

4 Bernstein Neural Network Framework Methodology

In this section, we present the application of the *Bernstein Neural Network* (BNN) framework to obtain the numerical solution of the fractional partial differential equation (FPDE) system given by equations (6)-(9). The BNN is employed to approximate the solution of a system of four coupled time-fractional PDEs, each involving Caputo fractional derivatives that model memory-dependent dynamics.

To effectively capture the temporal memory effects inherent in fractional-order systems, we utilize the L1 finite-difference approximation for the Caputo derivative. This scheme approximates the fractional derivative at a given time step by a weighted sum of previous function values. The weights, referred to as *Caputo weights*, are precomputed and reused throughout the training process to enhance computational efficiency.

The training domain is discretized into a set of collocation points, and the residuals of each equation are evaluated at these points to guide the network optimization.

The L1 approximation of the Caputo fractional derivative $D_t^\alpha u(t)$ at a discrete time $t_n = n\Delta t$ is given by:

$$D_t^\alpha u(t_n) \approx \sum_{k=1}^{n} w_k \left(u(t_{n-k+1}) - u(t_{n-k}) \right), \tag{11}$$

where the weights w_k are defined as:

$$w_k = \frac{(\Delta t)^{-\alpha}}{\Gamma(2-\alpha)} \left[k^{1-\alpha} - (k-1)^{1-\alpha} \right],$$

with $\alpha \in (0,1)$ denoting the fractional order and $\Gamma(\cdot)$ representing the Gamma function.

In the framework of the Bernstein Neural Network (BNN), the unknown functions $N(x,t)$, $T(x,t)$, $I(x,t)$, and $U(x,t)$ representing the densities of normal cells, tumor cells, immune cells, and chemotherapeutic drug concentration, respectively are approximated using a tensor product of Bernstein basis polynomials. This approach leverages the spectral approximation capability of Bernstein polynomials, which are well-known for their stability, positivity, and shape-preserving properties in approximation theory [**?**, **?**].

The two-dimensional Bernstein polynomial expansion is constructed over the spatial and temporal domains using the normalized variables x' and t', which map the original domains to the unit interval $[0,1]$. The approximations are given by:

$$\begin{aligned}
N(x,t) &\approx \sum_{i=0}^{n}\sum_{j=0}^{n} p_{ij}\, B_{i,n}(x')\, B_{j,n}(t'),\\
T(x,t) &\approx \sum_{i=0}^{n}\sum_{j=0}^{n} q_{ij}\, B_{i,n}(x')\, B_{j,n}(t'),\\
I(x,t) &\approx \sum_{i=0}^{n}\sum_{j=0}^{n} r_{ij}\, B_{i,n}(x')\, B_{j,n}(t'),\\
U(x,t) &\approx \sum_{i=0}^{n}\sum_{j=0}^{n} s_{ij}\, B_{i,n}(x')\, B_{j,n}(t'),
\end{aligned}$$

where the normalized coordinates are defined as:

$$x' = \frac{x - x_{\min}}{x_{\max} - x_{\min}}, \qquad t' = \frac{t - t_{\min}}{t_{\max} - t_{\min}}.$$

Here, $B_{i,n}(\cdot)$ denotes the Bernstein basis polynomial of degree n, and the coefficients $p_{ij}, q_{ij}, r_{ij}, s_{ij}$ serve as the trainable weights of the BNN, to be optimized during the learning process. This representation facilitates the construction of a differentiable function class that satisfies the structure of fractional partial differential equations.

The BNN framework effectively integrates the polynomial approximation properties with data-driven optimization, enabling accurate approximation of the solutions even in the presence of memory-dependent (nonlocal) dynamics governed by fractional derivatives. Furthermore, due to the global smoothness of Bernstein basis functions and their ability to handle boundary constraints naturally, this method is particularly advantageous for solving biologically inspired FPDEs with complex initial-boundary conditions.

4.1 Bernstein Polynomial-Based Feature Construction for Neural Network Input

To effectively incorporate the structure-preserving properties of Bernstein polynomials into the neural network framework, we define an input feature transformation based on the Bernstein basis functions. The input variables (x, t), corresponding to spatial and temporal coordinates, are first rescaled to the unit square $[0, 1] \times [0, 1]$. The rescaled variables are given by:

$$x' = \frac{x - x_{\min}}{x_{\max} - x_{\min}}, \qquad t' = \frac{t - t_{\min}}{t_{\max} - t_{\min}}.$$

These normalized inputs are then transformed into a high-dimensional feature vector $\mathbf{F}(x, t) \in \mathbb{R}^{2n+4}$ using the tensor products of one-dimensional Bernstein basis functions of degree n. The feature vector incorporates both the original variables and their Bernstein representations, and is constructed as:

$$\mathbf{F}(x,t) = \begin{bmatrix} x \\ t \\ B_{0,n}(x') \\ B_{1,n}(x') \\ \vdots \\ B_{n,n}(x') \\ B_{0,n}(t') \\ B_{1,n}(t') \\ \vdots \\ B_{n,n}(t') \end{bmatrix} \in \mathbb{R}^{2+2(n+1)} = \mathbb{R}^{2n+4}.$$

This vector serves as the input to the neural network and captures both local and global features of the spatial-temporal domain. The use of Bernstein basis functions ensures smoothness, boundary adherence, and numerical stability, which are essential in solving fractional partial differential equations. By embedding such a feature structure, the Bernstein Neural Network (BNN) gains enhanced representational capacity and is better suited for approximating the nonlocal dynamics governed by time-fractional operators.

4.2 Neural Network Architecture and Output Representation

The Bernstein-based feature vector $\mathbf{F}(x, t) \in \mathbb{R}^{2n+4}$, constructed as described in the previous section, serves as the input $\mathbf{h}_0$ to a fully connected feedforward neural network. The architecture consists of three hidden layers with hyperbolic tangent (tanh) activation functions, followed by a final linear output layer.

The hidden layer transformations are given by:

$$\begin{aligned}\mathbf{h}_1 &= \tanh(\mathbf{W}_1\mathbf{h}_0 + \mathbf{b}_1),\\ \mathbf{h}_2 &= \tanh(\mathbf{W}_2\mathbf{h}_1 + \mathbf{b}_2),\\ \mathbf{h}_3 &= \tanh(\mathbf{W}_3\mathbf{h}_2 + \mathbf{b}_3),\end{aligned}$$

and the output layer is defined as:

$$\mathbf{y} = \mathbf{W}_4\mathbf{h}_3 + \mathbf{b}_4,$$

where the output $\mathbf{y} \in \mathbb{R}^m$ provides predictions for the set of dependent variables $[v_1 = N(x,t), v_2 = T(x,t), v_3 = I(x,t), v_4 = U(x,t)]^T$.

Here, $\mathbf{W}_i$ and $\mathbf{b}_i$ represent the weight matrices and bias vectors of the respective layers. Specifically:

- $\mathbf{W}_1 \in \mathbb{R}^{d_1 \times d_0}$, $\mathbf{b}_1 \in \mathbb{R}^{d_1}$
- $\mathbf{W}_2 \in \mathbb{R}^{d_2 \times d_1}$, $\mathbf{b}_2 \in \mathbb{R}^{d_2}$
- $\mathbf{W}_3 \in \mathbb{R}^{d_3 \times d_2}$, $\mathbf{b}_3 \in \mathbb{R}^{d_3}$
- $\mathbf{W}_4 \in \mathbb{R}^{m \times d_3}$, $\mathbf{b}_4 \in \mathbb{R}^m$

The dimensions are defined as follows:

$d_0 = 2n + 4$ (input dimension)
$d_1 =$ number of neurons in the first hidden layer
$d_2 =$ number of neurons in the second hidden layer
$d_3 =$ number of neurons in the third hidden layer
$m =$ number of output variables

Each output component $v_i(x,t)$ can be explicitly expressed as:

$$v_i(x,t) = \mathbf{w}_i^T\mathbf{h}_3 + r_i, \quad \text{for } i = 1, 2, \ldots, m,$$

where $\mathbf{w}_i^T$ is the i-th row of the weight matrix $\mathbf{W}_4$, and r_i is the i-th component of the bias vector $\mathbf{b}_4$.

4.3 Training Strategy and Loss Function Formulation

The proposed neural network architecture is designed to learn complex, nonlinear mappings between the spatial-temporal features and the output variables governed by the system of coupled fractional partial differential equations (FPDEs).

The trainable parameters, including the weight matrices and bias vectors, are initialized using the **Xavier initialization** scheme to ensure stable gradient flow at the beginning of training. The optimization process is carried out using the **Adam optimizer**, which combines the benefits of adaptive learning rates and momentum for efficient convergence.

The total loss function is composed of several mean squared error (MSE) terms corresponding to:

1. Residuals of the FPDE system,
2. Initial condition constraints,
3. Boundary condition constraints.

The complete loss function is defined as:

$$\mathcal{L} = \sum_{k=1}^{m} \text{MSE}(\text{Res}_k) + \text{MSE}(\text{IC}) + \text{MSE}(\text{BC}),$$

where:

- $\text{MSE}(\text{Res}_k)$ penalizes the violation of the k^{th} equation residual at interior points,
- MSE(IC) enforces adherence to the known initial conditions,
- MSE(BC) enforces satisfaction of the prescribed boundary conditions.

Each of these components is explicitly defined as follows:

1. Residual Loss:

$$\text{MSE}(\text{Res}_k) = \frac{1}{N_r} \sum_{i=1}^{N_r} \left(\text{Res}_k(x_i, t_i)\right)^2,$$

where $\text{Res}_k(x_i, t_i)$ denotes the residual of the k^{th} FPDE evaluated at the interior collocation points (x_i, t_i), and N_r is the total number of such residual points. Residuals are computed using a combination of automatic differentiation (for classical terms) and finite-difference approximations (for fractional terms).

2. Initial Condition Loss:

$$\text{MSE}(\text{IC}) = \frac{1}{N_0} \sum_{i=1}^{N_0} \sum_{k=1}^{m} \left(u_k(x_i, 0) - u_k^{(0)}(x_i)\right)^2,$$

where $u_k^{(0)}(x_i)$ is the known initial condition for the k^{th} state variable, and N_0 is the number of initial condition sampling points.

3. Boundary Condition Loss:

$$\text{MSE}(\text{BC}) = \frac{1}{N_b} \sum_{i=1}^{N_b} \sum_{k=1}^{m} \left(u_k(x_i^{\text{BC}}, t_i^{\text{BC}}) - g_k(x_i^{\text{BC}}, t_i^{\text{BC}})\right)^2,$$

where g_k is the prescribed boundary function for the k^{th} output variable, and $(x_i^{\text{BC}}, t_i^{\text{BC}})$ are sampled boundary points with a total count of N_b.

Training Procedure: During training, the loss function $\mathcal{L}$ is minimized by iteratively updating the network parameters through backpropagation. At each iteration, forward propagation is used to compute the predicted outputs and residuals, followed by gradient computation and parameter updates via:

$$\mathbf{W}_i \leftarrow \mathbf{W}_i - \eta \frac{\partial \mathcal{L}}{\partial \mathbf{W}_i},$$

where η is the learning rate and $\frac{\partial \mathcal{L}}{\partial \mathbf{W}_i}$ is the gradient of the loss with respect to the weight matrix $\mathbf{W}_i$.

Model Evaluation: After training, the learned model is evaluated across temporal and spatial slices to analyze the system dynamics. Quantitative accuracy is assessed using standard error metrics such as:

- **Mean Absolute Error (MAE)**:

$$\text{MAE} = \frac{1}{N}\sum_{i=1}^{N}\left|u_i^{\text{pred}} - u_i^{\text{true}}\right|,$$

  where N is the number of evaluation points, u_i^{pred} is the predicted value, and u_i^{true} is the ground truth.

- **Root Mean Squared Error (RMSE)** and **Relative L^2 Norm Error** can also be computed to further validate model fidelity.

This comprehensive training approach ensures that the network solution satisfies the governing physics, initial conditions, and boundary constraints while maintaining numerical stability and generalization.

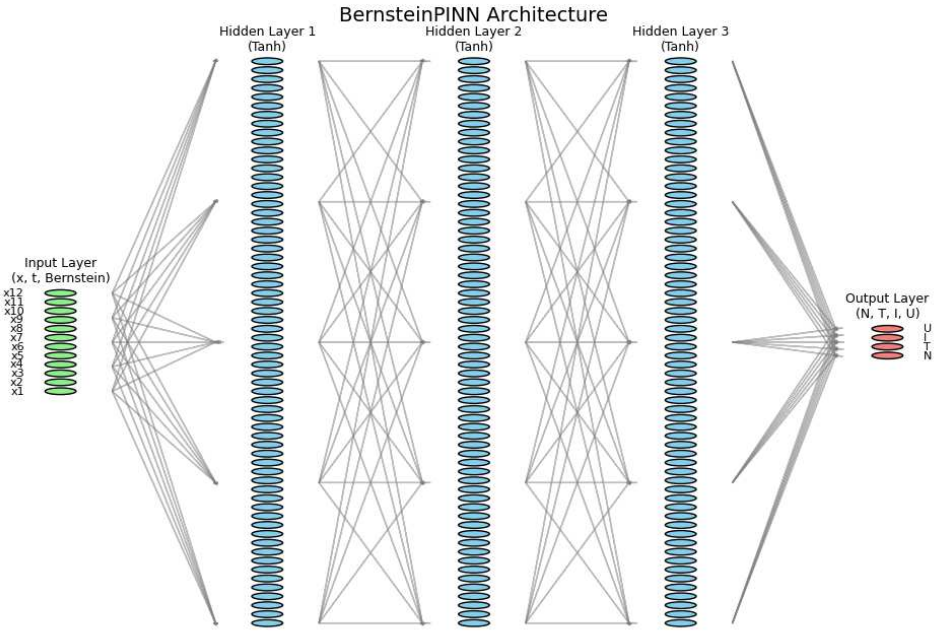


Figure 1: Architecture of the BernsteinPINN: input layer with Bernstein features, three hidden layers with tanh activations, and an output layer for N, T, I, U.

Algorithm: Bernstein Neural Network for four coupled FPDEs

Algorithm 1 Training and Evaluation of Bernstein Neural Network

Input: Domain $x \in [x_{\min}, x_{\max}]$, $t \in [t_{\min}, t_{\max}]$, grid sizes N_x, N_t, fractional orders $\{\alpha_k\}$, polynomial degree n, learning rate η, epochs E
Output: Trained model $\mathcal{N}_\theta$
procedure SETUP
  Define a four coupled FDE system and source terms $f_k(x,t)$
  Define Bernstein features $B_i(x'), B_j(t')$ of degree n
  Construct feature map $\mathbf{F}(x,t)$
  Build neural network: $\mathbf{F}(x,t) \to [u_1, \ldots, u_m]$
  Precompute L1 Caputo weights for each α_k
end procedure
procedure TRAIN AND EVALUATE
  for each α_k in $\{\alpha\}$ **do**
    Generate collocation points $\{(x_i, t_i)\}_{i=1}^{N_x N_t}$
    Compute forcing terms from exact solution if available
    Initialize model parameters θ and optimizer
    for epoch = 1 to E **do**
      Forward pass: predict $\hat{u}_k(x_i, t_i)$
      Compute PDE residuals using autograd + fractional L1
      Compute $\mathcal{L}$ from residuals, IC, BC
      Backpropagate and update θ
    end for
    Evaluate model on fixed $x = x_0$ and $t = t_0$
    Compute and store MAE, RMSE, and relative L^2 error
  end for
end procedure

5 Numerical Results and Discussion

In this section, we present numerical experiments to demonstrate the accuracy, efficiency, and applicability of the proposed Bernstein Neural Network (BNN) method for solving the time-fractional partial differential equation system defined in equations (6)–(9). The computational implementation has been carried out using the `Python` programming language.

Example 1: To validate the effectiveness of our proposed framework, we consider a specific instance of the model with the fractional order parameter set to $\alpha = 0.9$. The selected configuration represents a biologically relevant scenario within the tumor growth framework under chemotherapeutic effects.

To evaluate the performance and generalization capability of the BNN-based solver, the model is trained using the Adam optimizer over 5000 epochs for three different values of the fractional order: $\alpha = 0.6, 0.7, 0.8$. The accuracy of the method is quantified by computing the Mean Absolute Error (MAE) between the exact and predicted numerical solutions.

Figures 2, 3, 4, and 5 present the MAE results for each of the four state variables $N(x,t)$, $T(x,t)$, $I(x,t)$, and $U(x,t)$, respectively. These figures illustrate the conver-

gence behavior and highlight the model's capability to capture the underlying fractional dynamics.

Furthermore, Table 2 summarizes the MAE values for all state variables across the chosen values of α. The numerical evidence supports that the proposed approach achieves a satisfactory degree of accuracy and effectively handles the complexity of coupled time-fractional systems.

Table 2: Mean Absolute Error (MAE) for Each State Variable

α	N	T	I	U
0.6	1.226×10^{-3}	8.750×10^{-4}	6.740×10^{-4}	4.673×10^{-4}
0.7	1.743×10^{-3}	7.375×10^{-4}	3.831×10^{-4}	6.394×10^{-4}
0.8	9.536×10^{-4}	7.852×10^{-4}	1.168×10^{-4}	8.169×10^{-4}

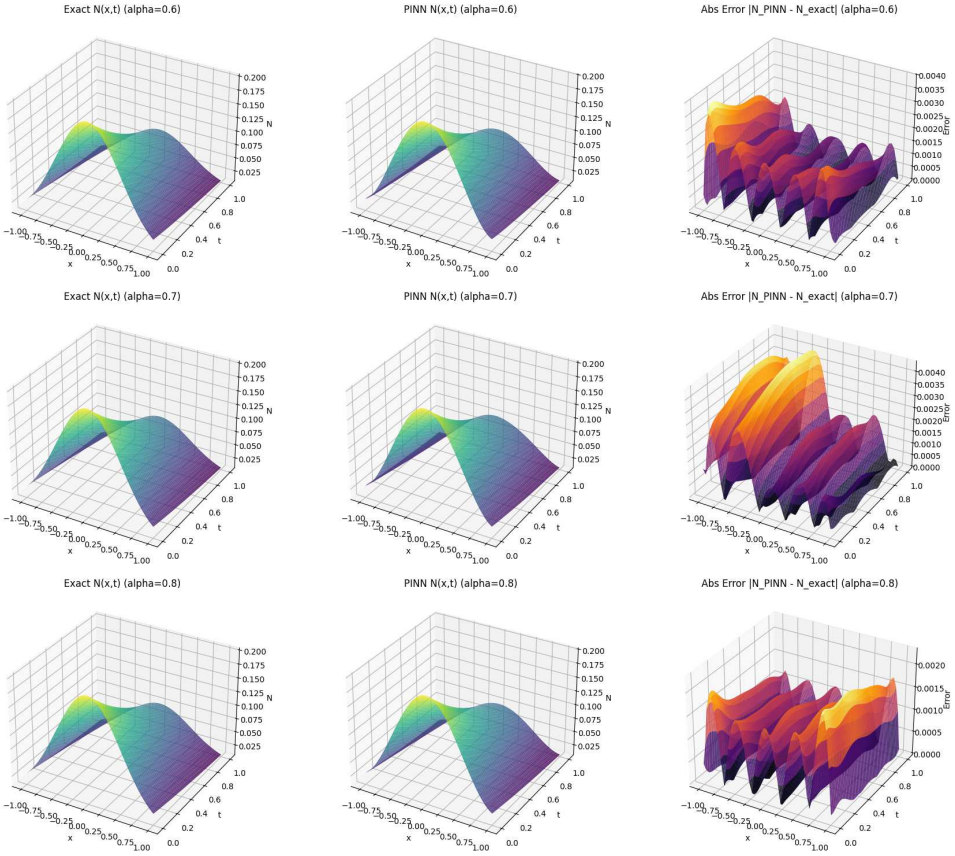


Figure 2: Plot of $N(x,t)$ for $\alpha = 0.6$, 0.7 and 0.8

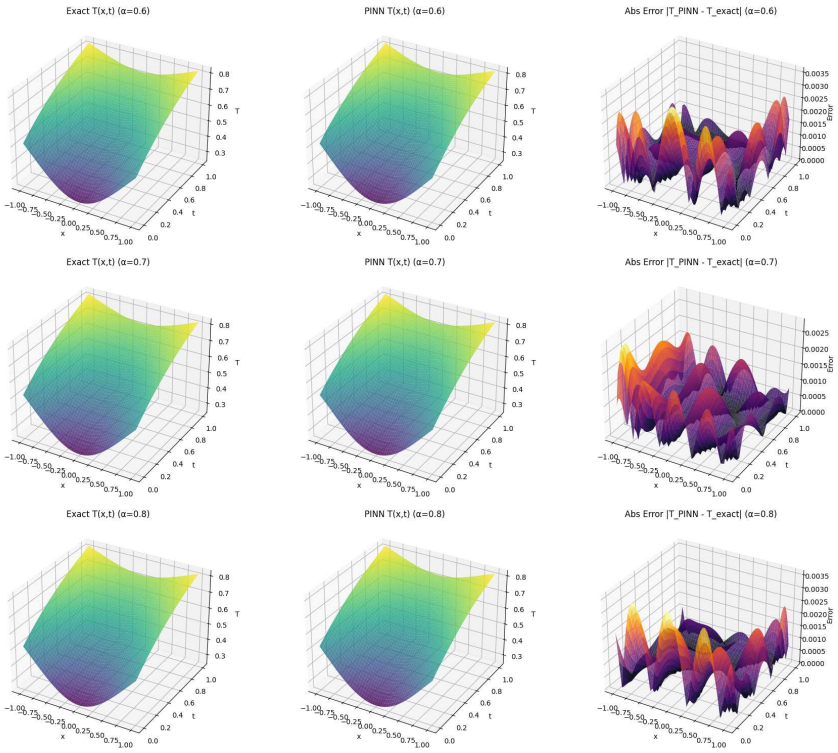


Figure 3: Plot of $T(x,t)$ for $\alpha = 0.6$, 0.7 and 0.8

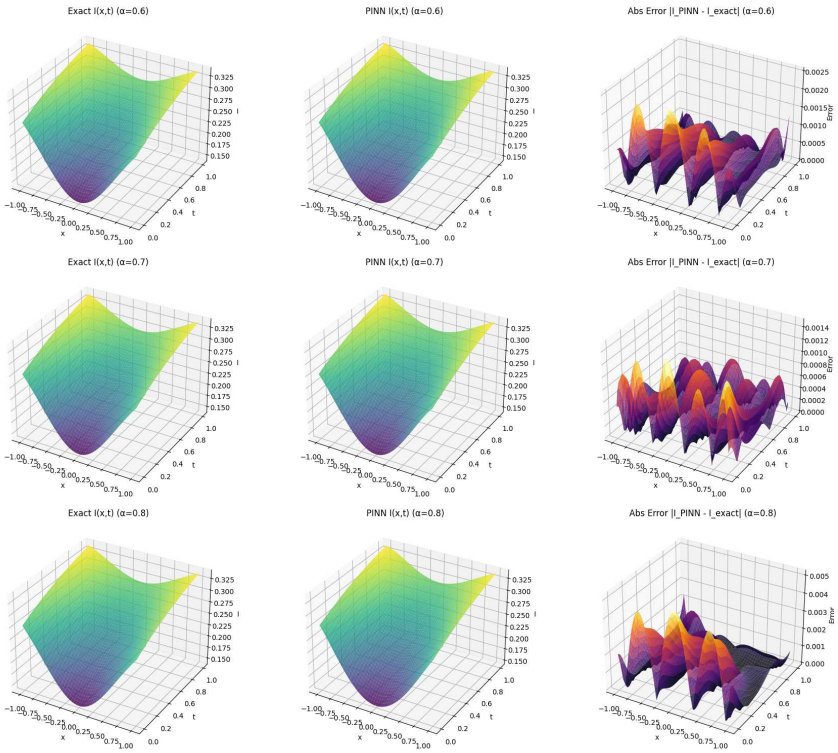


Figure 4: Plot of $U(x,t)$ for $\alpha = 0.6$, 0.7 and 0.8

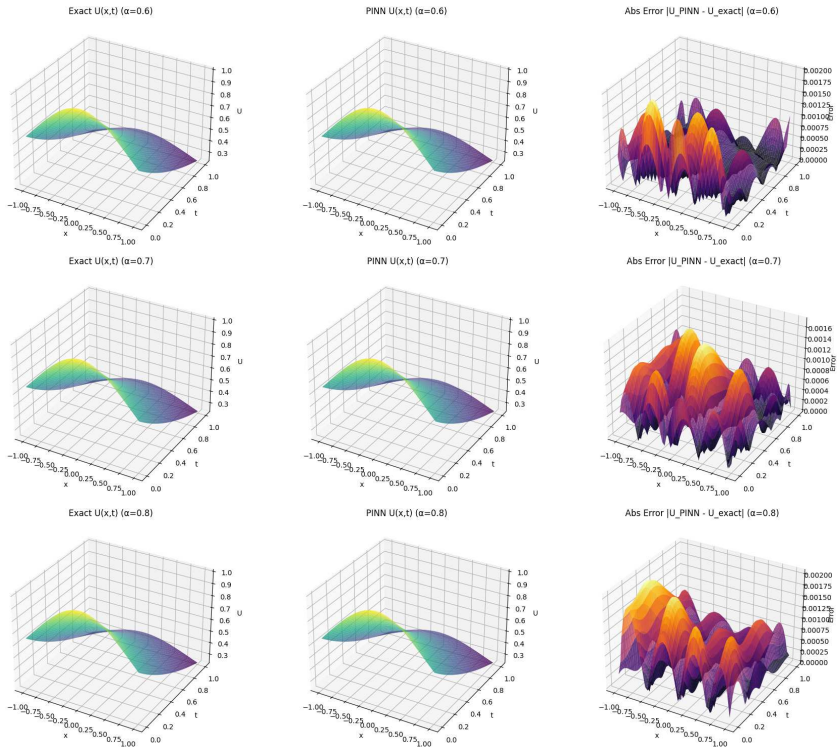


Figure 5: Plot of $I(x,t)$ for $\alpha = 0.6$, 0.7 and 0.8

5.1 Interpretation of Results for Normal Cell Concentration $N(x = 0.1, t)$ Under Varying Fractional Orders

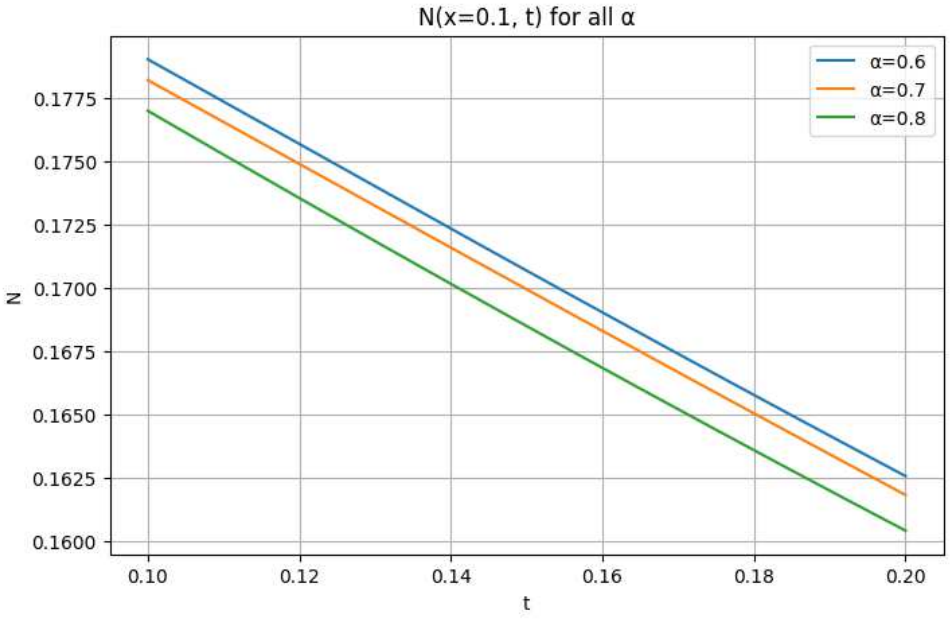


Figure 6: $N(x = 0.1, t)$ for $\alpha = 0.6$, 0.7 and 0.8

Figure 6 presents the temporal evolution of the normal cell concentration $N(x = 0.1, t)$ for different values of the fractional order $\alpha \in \{0.6, 0.7, 0.8\}$, over the time interval $t \in$

$[0.1, 0.2]$, within the context of a tumor-chemotherapy interaction model governed by a system of fractional differential equations.

The fractional derivative employed here reflects sub-diffusive transport phenomena, where the order $\alpha < 1$ captures the memory and hereditary effects intrinsic to biological tissues. As observed, increasing the fractional order from $\alpha = 0.6$ to $\alpha = 0.8$ results in a higher concentration of normal cells at each time point. This indicates that the normal cell population experiences less degradation or damage at higher fractional orders.

This behavior suggests that stronger memory effects (i.e., higher α) mitigate the rate of decline in normal cells, possibly due to reduced diffusion-driven interactions with cytotoxic agents or tumor cells. Consequently, the model implies that fractional-order dynamics can play a crucial role in preserving normal cell populations during chemotherapy, highlighting the importance of fractional modeling in understanding complex biological responses.

5.2 Interpretation of Results for Tumor Cell Concentration $T(x = 0.1, t)$ Under Varying Fractional Orders

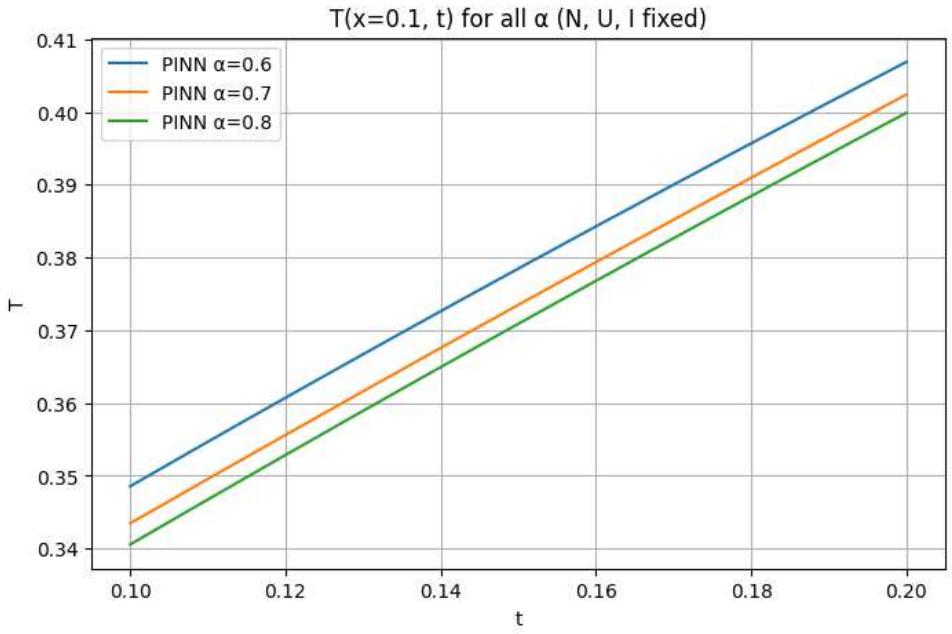


Figure 7: $T(x = 0.1, t)$ for $\alpha = 0.6$, 0.7 and 0.8

Figure 7 illustrates the temporal dynamics of the tumor cell concentration $T(x = 0.1, t)$ for three distinct values of the fractional order $\alpha \in \{0.6, 0.7, 0.8\}$ over the time interval $t \in [0.1, 0.2]$, based on a fractional-order tumor-chemotherapy interaction model.

The fractional derivative, which governs the diffusion process in the model, is subdiffusive in nature due to $\alpha < 1$. This reflects the non-local, memory-dependent behavior typical of biological systems. The simulation results indicate that the concentration of tumor cells increases more rapidly for lower values of α, whereas higher values of α (i.e., approaching classical diffusion) lead to a more controlled growth in tumor cell density.

This observed trend suggests that subdiffusive dynamics—more prominent at lower fractional orders—may limit the spatial spread and growth of tumor cells more effec-

tively. In contrast, as α increases (i.e., memory effects diminish), the tumor cells experience faster accumulation. Thus, the model highlights the crucial role of fractional-order diffusion in modulating tumor progression and potentially enhancing the effectiveness of chemotherapeutic intervention.

5.3 Interpretation of Results for Immune Cell Concentration $I(x = 0.1, t)$ Under Varying Fractional Orders

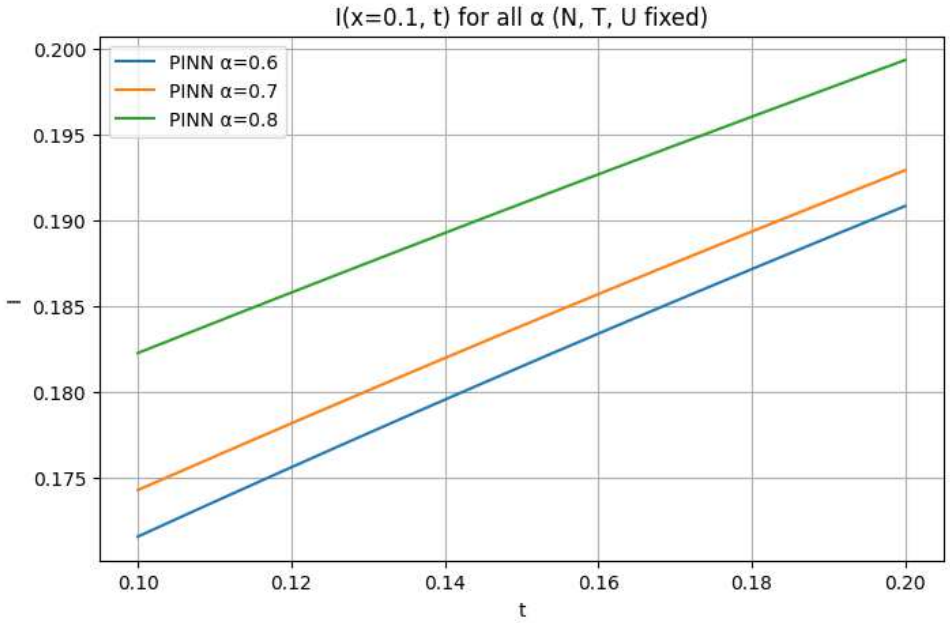


Figure 8: Temporal evolution of immune cell concentration $I(x = 0.1, t)$ for different fractional orders $\alpha \in \{0.6, 0.7, 0.8\}$.

Figure 8 presents the temporal evolution of the immune cell concentration $I(x = 0.1, t)$ for three values of the fractional order $\alpha \in \{0.6, 0.7, 0.8\}$ over the interval $t \in [0.1, 0.2]$, as determined by the underlying fractional-order tumor-immune-chemotherapy interaction model.

In this model, the Caputo-type fractional derivative governs the memory-dependent dynamics of immune cell transport and response. Due to $\alpha < 1$, the system exhibits subdiffusive behavior, capturing the delayed, history-sensitive activation and migration patterns commonly observed in immune processes.

The simulation results reveal that higher values of α correspond to more pronounced increases in immune cell concentration over time. Conversely, lower values of α (i.e., stronger memory effects) result in a slower buildup of immune presence at the spatial location $x = 0.1$. This behavior underscores the regulatory effect of fractional-order dynamics on immune activation—where reduced fractional order dampens the immune response temporally due to longer memory retention and slower effective diffusion.

Overall, this trend suggests that enhancing immune activation may be associated with higher fractional orders, reflecting reduced memory effects and more immediate cell mobilization. From a modeling perspective, the results emphasize the importance of fractional dynamics in shaping immune system behavior in response to tumor progression and therapy.

5.4 Interpretation of Results for Chemotherapeutic Drug Concentration $U(x = 0.1, t)$ Under Varying Fractional Orders

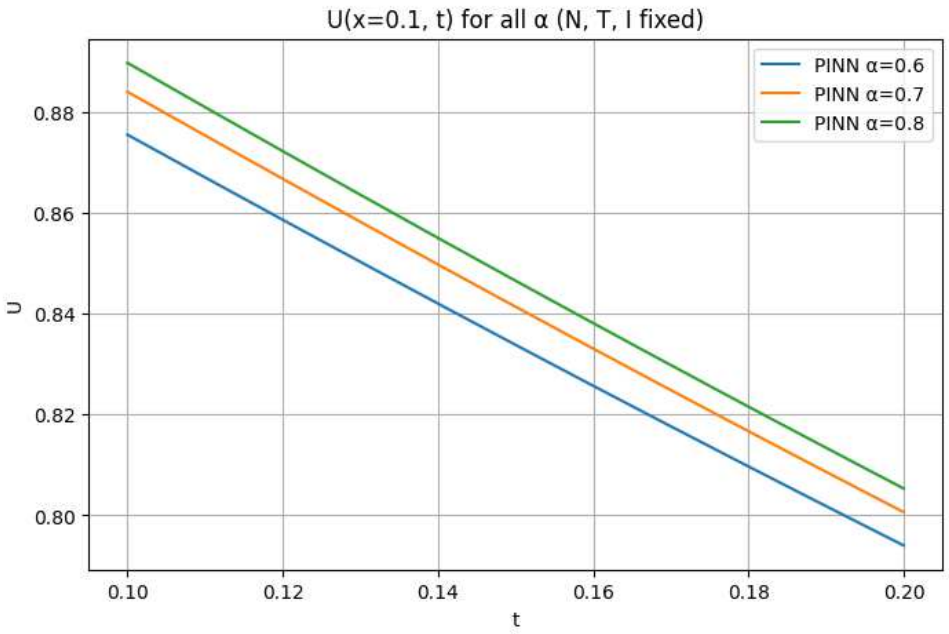


Figure 9: Temporal evolution of chemotherapeutic drug concentration $U(x = 0.1, t)$ for different fractional orders $\alpha \in \{0.6, 0.7, 0.8\}$.

Figure 9 illustrates the temporal dynamics of the chemotherapeutic agent concentration $U(x = 0.1, t)$ for three distinct values of the fractional order $\alpha \in \{0.6, 0.7, 0.8\}$, over the time interval $t \in [0.1, 0.2]$. These results are obtained from the fractional-order tumor-immune-chemotherapy interaction model, where the fractional dynamics govern the diffusion and decay behavior of the drug. The model incorporates a Caputo-type fractional derivative, which introduces memory effects into the temporal evolution of the drug concentration. For $\alpha < 1$, the dynamics are sub-diffusive, indicating that the transport and degradation of the drug are influenced by the system history, leading to slower-than-classical diffusion. Simulation results show that the concentration U consistently decreases over time for all values of α, reflecting natural drug decay or consumption. Notably, higher values of α correspond to slower decay rates, resulting in higher residual drug concentration at any given time. In contrast, lower values of α (i.e., stronger memory effects) accelerate the decay, leading to reduced drug presence over the same interval. This behavior suggests that memory-driven (subdiffusive) dynamics enhance drug dissipation in the modeled system. Consequently, tuning the fractional order α provides a mechanism to control the drug residence time and efficacy, with higher α promoting prolonged therapeutic availability. The results emphasize the role of fractional-order processes in shaping the pharmacokinetics of chemotherapeutic agents within a biologically realistic framework.

6 Discussion

In this research work, a fractional-order mathematical model representing the interaction of tumor cells under the influence of chemotherapeutic treatment has been inves-

tigated. To numerically solve the system of coupled time-fractional partial differential equations, a Bernstein polynomial-based neural network (BNN) framework was developed and applied. The numerical results demonstrate that the proposed BNN approach consistently achieves high accuracy across all tested values of the fractional order α. The method is particularly effective in capturing the inherent memory effects characteristic of fractional-order models, which are essential for accurately modeling real-world biological and physical systems. In comparison to conventional numerical schemes, Bernstein neural networks (PINNs) offer several distinct advantages: they are mesh-free, differentiable with respect to inputs and parameters, and well-suited to handling nonlinear, coupled systems and irregular geometries. The temporal and spatial dynamics of normal cells, tumor cells, and immune cells have been studied and visualized under different values of the fractional order α. These graphical results reinforce the effectiveness of the proposed approach in modeling and simulating complex biological systems governed by nonlocal, memory-driven behavior.

References

[1] I. Podlubny, *Fractional Differential Equations*. Academic Press, 1999.

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

[3] Unknown, "A comprehensive review of fractional calculus in science and engineering," *Current Advances in Natural Sciences*, 2025. March 6.

[4] S. Sellappan and E. Navajothiai, "Applications of fractional calculus in engineering," *Journal of Fractional Calculus and Applications*, pp. 1–20, 2022.

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

[6] C. Li and F. Zeng, *Numerical Methods for Fractional Calculus*. CRC Press, 2015.

[7] M. Hafez, F. Alshowaikh, B. W. N. Voon, S. Alkhazaleh, and H. Al-Faiz, "Review on recent advances in fractional differentiation and its applications," *Progress in Fractional Differentiation and Applications*, vol. 11, no. 2, pp. 245–261, 2025.

[8] D. Baleanu, Y. Karaca, L. Vazquez, and J. E. Macías-Díaz, "Advanced fractional calculus, differential equations and neural networks: analysis, modeling and numerical computations," *Physica Scripta*, vol. 98, 2023.

[9] F. Özköse, M. T. Şenel, and R. Habbireeh, "Fractional-order mathematical modelling of cancer cells-cancer stem cells-immune system interaction with chemotherapy," *Mathematical Modelling and Numerical Simulation with Applications*, vol. 1, no. 2, pp. 63–80, 2021.

[10] J. J. Nieto and A. Torres, "Fractional differential equations applied to tumor growth modeling," *Mathematical Modelling of Natural Phenomena*, vol. 11, no. 3, pp. 124–133, 2016.

[11] M. Zarei *et al.*, "Fractional-order models for cancer-immune interaction with memory effects," *Computational and Mathematical Methods in Medicine*, 2022.

[12] N. Bellomo, N. Li, and P. K. Maini, "Modeling tumor growth and immune response: A review," *Mathematical Models and Methods in Applied Sciences*, vol. 18, no. 4, pp. 593–646, 2008.

[13] M. N. Mishra, M. K. Singh, and A. K. Tiwari, "Mathematical modelling of growth of tumour cells with chemotherapy and immunotherapy," *Journal of Carcinogenesis & Mutagenesis*, vol. 13, no. 1, p. e2146572, 2022.

[14] P. Singh and D. Kumar, "Bernstein polynomial-based neural networks for solving fractional pdes," *Applied Mathematics and Computation*, 2022.

[15] H. Jafari *et al.*, "Neural network methods in fractional calculus and applications," *Mathematics*, 2021.

[16] N. M. Tuan *et al.*, "A neural network approach for solving fractional differential equations with bernstein polynomials," *Chaos, Solitons & Fractals*, 2020.

[17] K. M. Owolabi, "Riemann-liouville fractional derivative and application to model chaotic differential equations," *Progress in Fractional Differentiation and Applications*, vol. 4, no. 2, pp. 99–110, 2018.

[18] M. Caputo, "Linear models of dissipation whose q is almost frequency independent—ii," *Geophysical Journal International*, vol. 13, no. 5, pp. 529–539, 1967.

[19] S. Kumar and A. Atangana, "A numerical study of the nonlinear fractional mathematical model of tumor cells in presence of chemotherapeutic treatment," *International Journal of Biomathematics*, vol. 13, no. 03, p. 2050021, 2020.