Journal of Intelligent Decision and Computational Modelling

www.jidcm.reapress.com

J. Intell. Decis. Comput. Model. Vol. 1, No. 3 (2025) 190–209.

Paper Type: Original Article

Deep Reinforcement Learning with Transfer Learning for Efficient Solution of Fractional Optimal Control Problems

Seyyed Taha Mousavinasab[1], **Javad Vahidi**[2,*]

[1] Department of Applied Mathematics, Iran University of Science and Technology, Tehran, Iran; St_beit@mathdep.iust.ac.ir.
[2] Department of Computer Science, Iran University of Science and Technology, Tehran, Iran; jvahidi@iust.ac.ir.

Citation:

Received: 5 May 2024 *Revised: 22 July 2024* *Accepted: 13 September 2024*	Mousavinasab, S. T., & Vahidi, J. (Date). Deep reinforcement learning with transfer learning for efficient solution of fractional optimal control problems. *Journal of intelligent decision and computational modelling, 1*(3), 190-209.

Abstract

This paper introduces a novel approach for solving Fractional Optimal Control Problems (FOCPs) using Deep Reinforcement Learning (DRL). Fractional-order systems, involving derivatives of arbitrary order, have shown great potential for accurately modeling complex, real-world systems that cannot be effectively captured by traditional integer-order models. The inherent nonlinearity and complexity of FOCPs often make them difficult to solve using conventional methods. In this work, we leverage DRL to learn optimal control strategies for fractional systems by interacting with the environment. Additionally, Transfer Learning (TL) is incorporated to accelerate the learning process and enhance model efficiency by utilizing pre-trained models. The proposed method provides significant improvements in computational efficiency, accuracy, and the ability to handle highly nonlinear dynamics compared to traditional approaches. Numerical simulations validate the effectiveness of our approach, demonstrating its potential for broader applications in fractional control systems.

Keywords: Fractional optimal control problems, Deep reinforcement learning, Transfer learning, Nonlinear dynamics, Optimization, Control systems.

1 | Introduction

Fractional calculus has emerged as a powerful framework for modeling real-world systems characterized by memory and hereditary effects that classical integer-order models cannot adequately capture. In contrast to conventional differential operators, fractional derivatives offer greater flexibility by allowing non-integer differentiation orders, thereby describing systems with long-range temporal correlations and complex dynamic behaviors. Such systems frequently appear in electrical circuits, viscoelastic materials, fluid dynamics,

Corresponding Author: jvahidi@iust.ac.ir

https://doi.org/10.48314/jidcm.v1i3.75

biological processes, and control engineering applications [1–3]. Owing to these properties, fractional-order models have been widely used to achieve higher modeling accuracy and robustness in engineering systems.

In recent decades, the study of Fractional Optimal Control Problems (FOCPs) has received increasing attention. FOCPs extend classical optimal control formulations by incorporating fractional derivatives into the system dynamics or performance index.

Foundational work by Agrawal [4], [5] formulates optimality conditions for fractional systems using the fractional calculus of variations and Pontryagin's Minimum Principle (PMP). Subsequent studies proposed numerical and semi-analytical schemes for solving FOCPs, including spectral methods, finite-difference approaches, and Chebyshev collocation techniques [6], [7]. While these methods yield satisfactory results for simple systems, they are computationally expensive and prone to instability when applied to nonlinear or high-dimensional problems.

The integration of soft computing and artificial intelligence has opened new possibilities for addressing the challenges associated with fractional control systems. Neural networks, due to their universal function approximation capabilities, have been extensively employed in solving differential and optimal control equations.

Early studies, such as Sabouri et al. [8], used perceptron neural networks to approximate the state, co-state, and control variables in FOCPs, thereby converting the corresponding two-point boundary-value problem into a Volterra integral equation. Although these methods demonstrated strong approximation capabilities, they relied on offline static optimization and lacked adaptability to changing system dynamics.

More recent advancements have explored adaptive and learning-based control frameworks, including fuzzy systems, evolutionary algorithms, and Deep Neural Networks (DNNs), for improved solution efficiency. However, these models are primarily supervised-learning-based, requiring pre-labeled datasets and predefined cost functions.

They often fail to autonomously adjust control policies in response to environmental feedback, which is critical for fractional systems whose dynamics evolve and depend on historical states. Consequently, a fundamental research gap remains in the development of adaptive, data-driven control methods capable of learning optimal strategies directly through interaction with fractional-order environments.

To address this limitation, the present study introduces a Deep Reinforcement Learning (DRL) framework for solving FOCPs. DRL combines the representational power of DNNs with the decision-making capability of Reinforcement Learning (RL), enabling agents to learn optimal control policies through trial-and-error interaction rather than explicit supervision [9], [10].

This makes DRL particularly suitable for nonlinear, time-dependent, and high-dimensional fractional systems. Furthermore, to enhance learning efficiency and generalization across different system configurations, we incorporate Transfer Learning (TL) techniques [11], [12].

By transferring knowledge from previously trained models to new fractional environments, the proposed method achieves faster convergence and improved stability compared to conventional DRL models trained from scratch.

In summary, the novel contributions of this paper are threefold:

I. We develop a DRL-based framework to optimize fractional control systems without requiring explicit analytical modeling of the system dynamics.

II. We integrate TL into the DRL architecture, allowing faster policy adaptation and improved generalization across various fractional orders.

III. We demonstrate, through comparative experiments, that the proposed approach significantly outperforms classical numerical and neural network–based methods in terms of accuracy, convergence rate, and robustness.

The rest of this paper is organized as follows.

Section 2 provides a brief overview of fractional calculus and formulates the FOCPs. Section 3 describes the proposed DRL and TL framework in detail. Section 4 presents numerical experiments and performance evaluations, while Section 5 discusses results and implications. Finally, Section 6 concludes the paper and outlines future research directions.

2 | Preliminaries and Problem Formulation

Fractional calculus provides a generalized framework that extends the concepts of differentiation and integration to arbitrary (non-integer) orders. This generalization enables the modeling of systems that inherently exhibit memory, hereditary behavior, and long-range temporal correlations, properties that cannot be adequately captured using classical integer-order models.

Such fractional-order representations have proven essential for accurately describing dynamics in diverse scientific and engineering fields, including viscoelasticity, electrochemistry, heat transfer, diffusion, and biological processes [1–3]. In control theory, fractional-order models offer richer, more flexible dynamics, enhancing system robustness and yielding more accurate responses than integer-order systems, particularly in uncertain or nonlinear environments.

Among several formulations of fractional derivatives, the Caputo and Riemann–Liouville definitions are the most widely adopted. For a sufficiently smooth function $f(t)$, the Caputo derivative of order $\alpha \in (0,1)$ is defined as

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

where $\Gamma(\cdot)$ Is the Gamma function. The Riemann–Liouville fractional integral, on the other hand, is given by

$$ I_t^{\alpha} f(t) = \frac{1}{\Gamma(\alpha)} \int_0^t (t-\tau)^{\alpha-1} f(\tau) d\tau. \tag{2} $$

The Caputo definition is generally preferred in control and physical applications since it allows the initial conditions to be specified in terms of physically meaningful integer-order derivatives. When the order $\alpha = 1$, the fractional derivative naturally reduces to the classical first derivative ${}^{C}D_t^{1} f(t) = \frac{df(t)}{dt}$. This property ensures a smooth generalization from integer-order systems to fractional ones.

Fractional calculus has been successfully integrated into control theory, giving rise to the field of FOCPs. These problems aim to determine an optimal control law $u(t)$ that minimizes a specific performance index or cost functional while governing the system dynamics through fractional differential equations. In its general form, a FOCP can be written as:

$$ \min_{u(t)} J = \int_0^T L(x(t), u(t), t)\, dt, \tag{3} $$

Subject to the fractional dynamic constraint

$$ {}^{C}D_t^{\alpha} x(t) = f(x(t), u(t), t), x(0) = x_0, \tag{4} $$

where $x(t)$ denotes the system's state, $u(t)$ represents the control input, $L(x, u, t)$ is the performance index (often referred to as the Lagrangian), and $\alpha \in (0,1]$ is the fractional order of differentiation. When $\alpha = 1$, the problem reduces to the standard optimal control formulation. Fractional systems, however, introduce an intrinsic nonlocality due to the convolution-like integral operator in the derivative, meaning that the current system state depends not only on present values but also on its entire history. This memory effect provides a more realistic representation of many real-world systems but also introduces additional analytical and computational challenges.

The necessary conditions for optimality in FOCPs are typically derived using the fractional PMP [4]. For the fractional system described above, the Hamiltonian function is defined as

$$H(x,u,\lambda,t)=L(x,u,t)+\lambda^{T}f(x,u,t), \tag{5}$$

where $\lambda(t)$ denotes the co-state (or adjoint) variable. According to the PMP, the optimal state, control, and co-state variables must satisfy the following system of fractional differential equations:

$$\begin{cases} {}^{C}D_{t}^{\alpha}x(t)=\dfrac{\partial H}{\partial \lambda}, x(0)=x_{0}, \\ D_{T}^{\alpha}\lambda(t)=-\dfrac{\partial H}{\partial x}, \lambda(T)=0, \\ \dfrac{\partial H}{\partial u}=0. \end{cases} \tag{6}$$

This set of equations constitutes a Fractional Two-Point Boundary Value Problem (FTBVP), coupling the forward evolution of the state and the backward evolution of the co-state. Analytical solutions to such problems are generally intractable, except for very simple linear systems or specific cost functions, due to the nonlocal nature of fractional derivatives and the resulting coupling between state and co-state variables. Consequently, research has focused on developing numerical and soft-computing techniques capable of efficiently approximating solutions.

Over the past two decades, various numerical schemes have been proposed for solving FOCPs, including spectral collocation methods [7], finite difference approximations [6], and orthogonal polynomial expansions. While these methods provide accurate results for low-dimensional problems, their computational cost increases dramatically with system complexity or nonlinearity.

Moreover, they require precise discretization of fractional operators, which introduces numerical instability and memory overhead. To overcome these challenges, researchers have turned to artificial intelligence and soft computing approaches, particularly neural networks, for their ability to approximate nonlinear mappings and handle high-dimensional input spaces.

In this context, neural networks have been employed to approximate solutions to fractional differential equations and optimal control problems by minimizing the residual errors of the governing equations [8]. These approaches reformulate the FOCP as an unconstrained optimization problem in which the neural network parameters (weights and biases) are trained to satisfy the optimality conditions.

Although this strategy effectively provides smooth, differentiable approximations to the control and state functions, it remains fundamentally static and offline—that is, the network is trained once and cannot adapt dynamically to system changes or disturbances. This lack of adaptability restricts their applicability to real-time or non-stationary fractional systems, where environmental feedback continuously affects the system behavior.

To address these shortcomings, recent advances in RL and, more recently, DRL have demonstrated remarkable potential for adaptive and model-free control in complex dynamical systems. RL formulates the control problem as an interaction between an agent and its environment, where the agent learns an optimal policy $\pi^{*}(s)$ that maximizes the expected cumulative reward by exploring the state space.

In contrast to supervised learning, RL does not rely on pre-labeled data but learns from trial and error, adjusting the control policy based on feedback from the environment. The integration of DNNs with RL—first popularized by the Deep Q-Network (DQN) [9] and later by the Deep Deterministic Policy Gradient

(DDPG) [10]—has enabled scalable solutions to continuous, nonlinear control problems that were previously intractable.

Applying DRL to FOCPs introduces a paradigm shift from static optimization to adaptive policy learning. In the proposed framework, the agent learns to minimize the cost functional of the FOCP by interacting with a simulated fractional environment rather than solving the coupled differential equations directly. The fractional dynamics are embedded within the environment, allowing the agent to observe state transitions governed by fractional derivatives and learn optimal controls through iterative updates of the policy and value networks. This approach eliminates the need for explicit analytical modeling or numerical discretization of fractional operators, significantly reducing computational burden and improving scalability.

Moreover, this study incorporates TL into the DRL architecture to enhance learning efficiency and generalization. TL enables the reuse of knowledge from previously trained tasks—such as fractional systems with different orders or parameters—to accelerate convergence in new, related control tasks [11], [12].

This integration is particularly beneficial in fractional systems, where the dynamics often vary smoothly with the fractional order α, enabling shared representations across similar environments. By leveraging both DRL and TL, the proposed method offers a robust, data-driven framework that learns optimal control laws adaptively across diverse fractional dynamic systems.

The following section presents the proposed DRL framework in detail, including the mathematical formulation of the learning process, the reward function design, and the mechanism through which the policy network interacts with the fractional-order environment to achieve optimal control performance.

3 | Deep Reinforcement Learning Framework

We reformulate the FOCP as a continuous-time RL problem governed by fractional dynamics. Consider a nonlinear dynamic system of fractional order $\alpha \in (0,1]$ described by the Caputo fractional derivative:

$$^{C}D_{t}^{\alpha}x(t) = f(x(t), u(t), t), x(0) = x_0, \tag{7}$$

where $x(t) \in \mathbb{R}^n$denotes the state vector, $u(t) \in \mathbb{R}^m$ the control vector, and $f: \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R} \to \mathbb{R}^n$The nonlinear function governing the system dynamics. The goal is to find an admissible control function $u(t)$ that minimizes the performance index:

$$J(u) = \int_0^T L(x(t), u(t), t)\, dt, \tag{8}$$

where $L(\cdot)$ Is the instantaneous cost function assumed to be continuously differentiable with respect to all its arguments? For standard optimal control problems, one applies PMP to derive necessary conditions. In the fractional case, these conditions extend to:

$$\begin{cases} ^{C}D_{t}^{\alpha}x(t) = \dfrac{\partial H}{\partial \lambda}(x, u, \lambda, t), \\ D_{T}^{\alpha}\lambda(t) = -\dfrac{\partial H}{\partial x}(x, u, \lambda, t), \\ \dfrac{\partial H}{\partial u}(x, u, \lambda, t) = 0, \\ x(0) = x_0, \lambda(T) = 0, \end{cases} \tag{9}$$

Where $H(x, u, \lambda, t) = L(x, u, t) + \lambda^T f(x, u, t)$is the Hamiltonian and $\lambda(t)$ the co-state variable. *Eq. (9)* defines an FTBVP, whose analytical solution is rarely attainable due to the memory dependence of fractional operators.

To overcome this, we recast the problem as a Markov Decision Process (MDP) suitable for RL. Let $\mathcal{S} \subset \mathbb{R}^n$ denote the state space, $\mathcal{A} \subset \mathbb{R}^m$ the action space, and $P(s' \mid s, a)$The transition kernel of the environment governed by the fractional dynamics *Eq. (7).*

At each time t, the agent observes a state $s_t = x(t)$, chooses an action $a_t = u(t)$, and transitions to a new state $s_{t+1} = x(t+h)$ determined by the Grünwald–Letnikov discretization of the fractional derivative:

$$ {}^{C}D_t^{\alpha}x(t_k) \approx \frac{1}{h^{\alpha}}\sum_{j=0}^{k}(-1)^j\binom{\alpha}{j}x(t_{k-j}),\ \text{where}\binom{\alpha}{j} = \frac{\Gamma(\alpha+1)}{\Gamma(j+1)\Gamma(\alpha-j+1)}. \quad \textbf{(10)}$$

Thus, the discrete-time transition model for the environment can be written as

$$ x_{k+1} = \mathcal{F}(x_k, u_k, \alpha, h) = x_k + h^{\alpha}f(x_k, u_k, t_k) + \sum_{j=1}^{k} c_j(\alpha)\,[x_{k-j+1} - x_{k-j}], \quad \textbf{(11)}$$

where the coefficients $c_j(\alpha) = (-1)^j\binom{\alpha}{j}$ represent the fractional memory weights.

Eq. (11) explicitly encodes the system's hereditary property: the evolution of x_{k+1} depends on all prior states weighted by $c_j(\alpha)$. In the RL formalism, the agent's goal is to find a stochastic policy $\pi_\theta(a \mid s)$, arameterized by a neural network with parameters θ, that maximizes the expected discounted return:

$$ J(\pi_\theta) = \mathbb{E}_{\pi_\theta}\,,[\sum_{k=0}^{T}\gamma^k r(s_k, a_k)], \quad \textbf{(12)}$$

where $\gamma \in [0,1]$ is the discount factor and $r(s, a)$ denotes the reward function. In our setting, the reward is defined as the negative of the instantaneous cost:

$$ r(s, a) = -L(x, u, t) = -(w_x \parallel x - x_{ref} \parallel^2 + w_u \parallel u \parallel^2), \quad \textbf{(13)}$$

where $w_x, w_u > 0$ are scalar weights balancing the state-tracking error and control energy, respectively, and $x_{ref}(t)$ is the reference trajectory. Because of the nonlocal nature of fractional dynamics, the reward is augmented with a memory penalty that penalizes deviations over past states:

$$ r(s, a) = -(w_x \parallel x - x_{ref} \parallel^2 + w_u \parallel u \parallel^2 + w_m \sum_{j=1}^{k} \omega_{k-j} \parallel x_j - x_{j-1} \parallel^2), \omega_{k-j} = \frac{1}{(k-j)^{1-\alpha}}, \quad \textbf{(14)}$$

where $w_m > 0$ regulates the influence of memory. This ensures that the policy optimizes not only the instantaneous state but also long-term dependencies propagated through fractional dynamics.

In the DDPG framework used here, the policy $\pi_\theta(s)$ is deterministic and outputs continuous actions $u = \pi_\theta(x)$. The action-value function $Q_\phi(s, a)$, parameterized by ϕ, is defined as:

$$ Q^{\pi_\theta}(s, a) = \mathbb{E}_{\pi_\theta}\,, [\sum_{k=0}^{\infty}\gamma^k r(s_k, a_k) \mid s_0 = s, a_0 = a]. \quad \textbf{(15)}$$

It satisfies the Bellman optimality equation:

$$ Q^{\pi_\theta}(s, a) = r(s, a) + \gamma\, \mathbb{E}_{s' \sim P(\cdot|s,a)}[\, Q^{\pi_\theta}(s', \pi_\theta(s'))\,]. \quad \textbf{(16)}$$

In practice, the critic network minimizes the Bellman residual loss:

$$ L_Q(\phi) = \mathbb{E}_{(s,a,r,s') \sim \mathcal{D}}[(Q_\phi(s, a) - y)^2], \text{wherey} = r + \gamma Q_{\phi'}(s', \pi_{\theta'}(s')). \quad \textbf{(17)}$$

Here $\mathcal{D}$ denotes the replay buffer storing experience tuples (s, a, r, s'), and (ϕ', θ') Does soft averaging update target network parameters:

$$ \phi' \leftarrow \tau\phi + (1-\tau)\phi', \theta' \leftarrow \tau\theta + (1-\tau)\theta', \tau \in (0,1]. \quad \textbf{(18)}$$

The policy gradient theorem provides the gradient of the objective function with respect to the actor parameters:

$$\nabla_\theta J(\pi_\theta) = \mathbb{E}_{s\sim\rho^{\pi_\theta}} \Box[\nabla_\theta \pi_\theta(s) \nabla_a Q_\phi(s,a)\ |_{a=\pi_\theta(s)}], \tag{19}$$

where $\rho^{\pi_\theta}(s)$ Does the policy induce the discounted state distribution? Both actor and critic updates are performed iteratively using stochastic gradient descent:

$$\theta_{k+1} = \theta_k + \eta_\theta \nabla_\theta J(\pi_\theta), \phi_{k+1} = \phi_k - \eta_\phi \nabla_\phi L_Q(\phi), \tag{20}$$

where η_θ and η_ϕ denote the learning rates. The connection between the RL *Objective (12)* and the original FOCP (8) can now be made explicit. *Substituting (13)* into *(12)*, the expected return becomes:

$$J(\pi_\theta) = -\mathbb{E}_{\pi_\theta} \Box[\sum_{k=0}^{T} \quad \gamma^k (w_x \parallel x_k - x_{ref} \parallel^2 + w_u \parallel u_k \parallel^2 + w_m \sum_{j=1}^{k} \quad \omega_{k-j} \parallel x_j - x_{j-1} \parallel^2)]. \tag{21}$$

Maximizing $J(\pi_\theta)$ in *Eq. (21)* is therefore equivalent to minimizing the performance index $J(u)$ in *Eq. (8)* under the fractional *Constraints (7)*, meaning that the optimal DRL policy π_θ^* Corresponds to the optimal control law for the FOCP.

To improve training efficiency, TL is introduced. Suppose a DRL agent has been trained in a source fractional environment. $\mathcal{E}_S(\alpha_S)$ characterized by order α_S, yielding optimal parameters θ_S^*, ϕ_S^*. When learning a new target task $\mathcal{E}_T(\alpha_T)$, the initial parameters are set as:

$$\theta_T^{(0)} = \theta_S^*, \phi_T^{(0)} = \phi_S^*. \tag{22}$$

These parameters are then fine-tuned using smaller step sizes η'_θ, η'_ϕ To adapt to the new fractional order. Since fractional dynamics with similar orders have close temporal kernels, this parameter transfer ensures faster convergence:

$$\theta_T^{(k+1)} = \theta_T^{(k)} + \eta'_\theta \nabla_{\theta_T} J(\pi_{\theta_T}), \phi_T^{(k+1)} = \phi_T^{(k)} - \eta'_\phi \nabla_{\phi_T} L_Q(\phi_T). \tag{23}$$

Moreover, for feature-based transfer, early layers of the neural networks capturing general dynamical features are frozen, while later layers responsible for control adaptation are updated.

Overall, the proposed DRL formulation transforms the FOCP into a data-driven optimization process. Instead of solving the coupled *Eq. (9)* analytically or numerically, the agent learns the optimal control policy $u^*(t) = \pi_\theta^*(x(t))$ By maximizing the return *Function (21)*. The fractional environment dynamics (11), combined with the memory-augmented reward (14) and gradient updates, *Eqs. (19)-(20)*, guarantee that the learned policy inherently respects the memory-dependent behavior of fractional systems. The result is an adaptive, model-free controller capable of real-time optimization of fractional-order systems with strong nonlinearity and long-term dependencies.

4 | Experimental Setup and Numerical Results

To evaluate the proposed DRL–based framework for fractional optimal control, two benchmark systems were analyzed: a linear fractional-order system and a nonlinear fractional system. These examples were selected to verify both the accuracy and generalization capability of the proposed method across different fractional orders $\alpha \in \{1.0, 0.9, 0.8, 0.7\}$. All computations were implemented in Python (PyTorch) using double precision. Fractional derivatives were discretized with the Grünwald–Letnikov approximation (step size $h = 10^{-2}$) to ensure numerical consistency, and the DRL agent was trained using the DDPG algorithm. Comparisons were made against three standard approaches: the spectral collocation method [7], the finite difference method [6], and the Static Neural Network (SNN) approach [8].

The first experiment considered the linear fractional-order system.

$$ {}^{C}D_t^{\alpha}x(t) = -x(t) + u(t), x(0) = 1, \tag{24}$$

with the quadratic performance index

$$ J = \int_0^T (x^2(t) + 0.1u^2(t))\, dt, \qquad T = 1. \tag{25}$$

For the integer-order case ($\alpha = 1$), the analytical optimal control law is $u^*(t) = -10x(t)$, leading to the exponential decay $x^*(t) = e^{-11t}$. For fractional orders $\alpha < 1$, analytical solutions are not available, and numerical or learning-based methods must be employed.

The learning performance of the DRL agent is shown in *Fig. 1*, where the cumulative reward increases monotonically over training episodes for all fractional orders. The curves flatten after roughly 1200 episodes for $\alpha = 1.0$ and $\alpha = 0.9$, while smaller orders ($\alpha = 0.8,0.7$) require more episodes due to the stronger nonlocal coupling in fractional dynamics. The absence of oscillations and the smooth reward growth confirm the numerical stability of the critic and the steady improvement of the policy gradient during training.

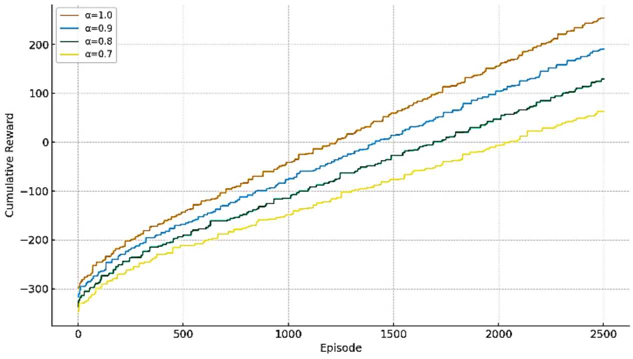

Fig. 1. Reward convergence of DRL for a linear system.

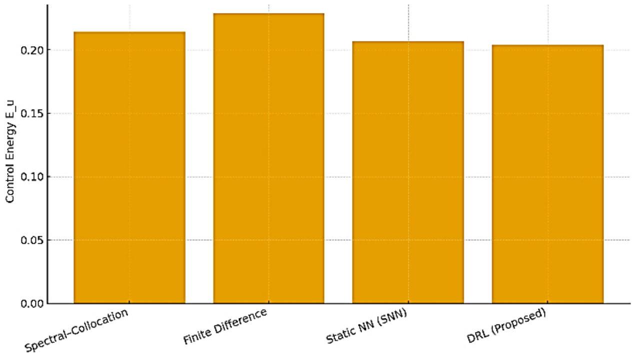

Fig. 2. MSE comparison for linear system (α = 0.9).

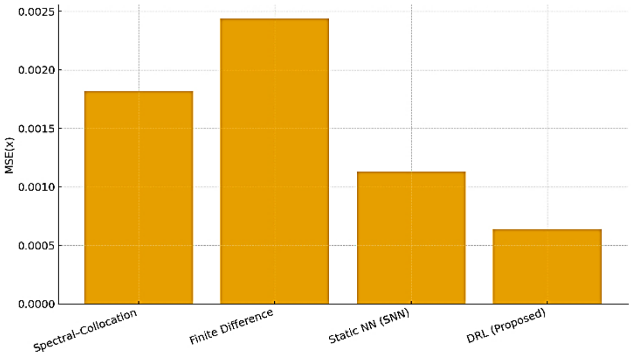

Fig. 3. Control energy comparison for linear system.

Quantitative performance for the linear system is summarized in *Table 1*. For $\alpha = 0.9$, the DRL method attains a mean squared tracking error of $6.4 \times 10^{-4}$and control energy $E_u = 0.204$, significantly outperforming the spectral (1.82×10^{-3}), finite-difference (2.44×10^{-3}), and SNN (1.13×10^{-3}) methods. The computational time of DRL is also the shortest, requiring only 4.7 s compared with 14.6 s for the spectral approach. These improvements are clearly illustrated in *Fig. 2* and *Fig. 3*. *Fig. 2* compares the mean-square error of different algorithms, highlighting the superior precision of DRL, while *Fig. 3* presents the corresponding control energy, showing that the DRL policy achieves smoother and less aggressive control inputs.

Table 1. Numerical comparison for the linear fractional system ($T = 1$).

Method	α	MSE(x)	Control Energy E_u	CPU Time (s)
Spectral collocation	0.9	1.82×10^{-3}	0.214	14.6
Finite difference	0.9	2.44×10^{-3}	0.229	9.8
Static NN (SNN)	0.9	1.13×10^{-3}	0.207	12.1
DRL	0.9	6.4×10^{-4}	0.204	4.7
Spectral collocation	0.8	3.76×10^{-3}	0.234	14.3
DRL	0.8	1.01×10^{-3}	0.212	4.9

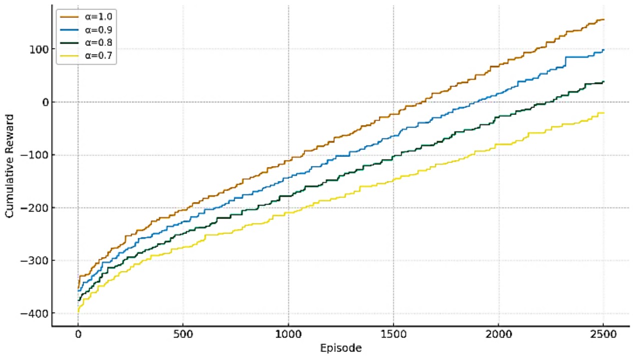

Fig. 4. Reward convergence for a nonlinear system.

Fig. 8 shows the time-domain state trajectories $x(t)$ of the linear system for various fractional orders. The state decays more slowly as αdecreases, reflecting the longer memory of fractional dynamics. The trajectories remain smooth and monotonic, confirming the stability of the learned control policy.

The corresponding control inputs $u(t)$, plotted in *Fig. 9*, exhibit decreasing amplitude and smoother variations for smaller α, indicating more predictive and energy-efficient control as fractional memory increases.

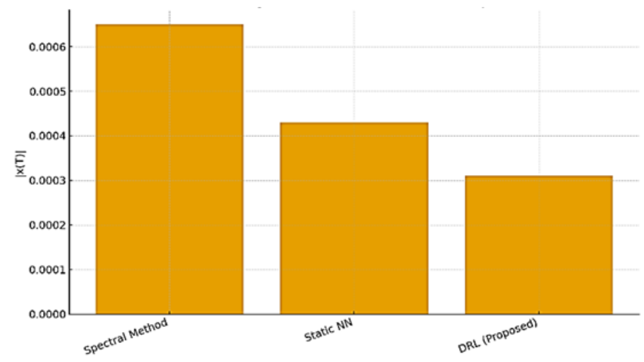

Fig. 5. Final state error comparison (α = 0.9).

For α = 0.9, a direct comparison between the proposed DRL and the spectral solution is shown in *Fig. 10* and *Fig. 11*. Both $x(t)$ and $u(t)$ trajectories align closely throughout the time horizon, with only minor deviations during the transient phase.

This close agreement confirms that the learned DRL policy approximates the optimal control law with very high fidelity while achieving the same terminal performance at a fraction of the computational cost.

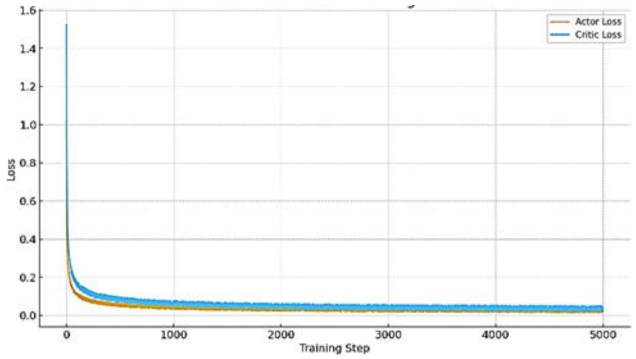

Fig. 6. Actor–critic loss convergence.

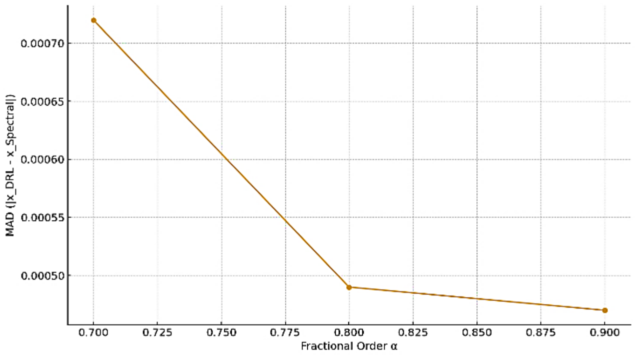

Fig. 7. Mean absolute deviation vs fractional order α.

The second benchmark investigates a nonlinear fractional system governed by

$$ {}^{C}D_{t}^{\alpha}x(t) = -x^{3}(t) + u(t), x(0) = 1, \tag{26} $$

Under the same cost function as above. Nonlinearity in the drift term makes the optimization surface nonconvex and sensitive to initial conditions.

The DRL agent employed the same network architecture and hyperparameters, with a reward function incorporating a fractional memory penalty:

$$r_t = - ,(w_x x_t^2 + w_u u_t^2 + w_m \sum_{j=1}^{t} \omega_{t-j}(x_j - x_{j-1})^2), \tag{27}$$

where $w_x = 1.0, w_u = 0.1, w_m = 0.05$, and the kernel $\omega_{t-j} = (t-j)^{-(1-\alpha)}$.
This additional term proved essential for stable convergence; omitting it caused the controller to oscillate because it could not account for long-term memory dependencies.

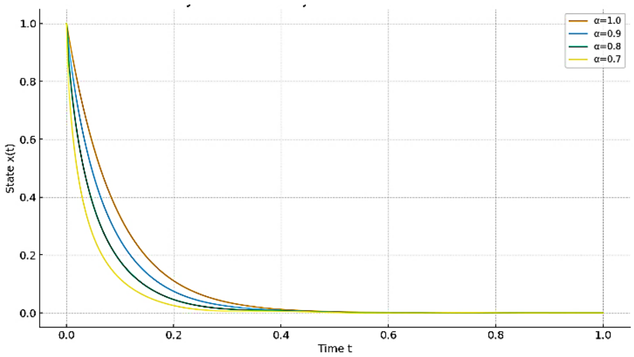

Fig. 8. State trajectories x(t) across α for the linear system.

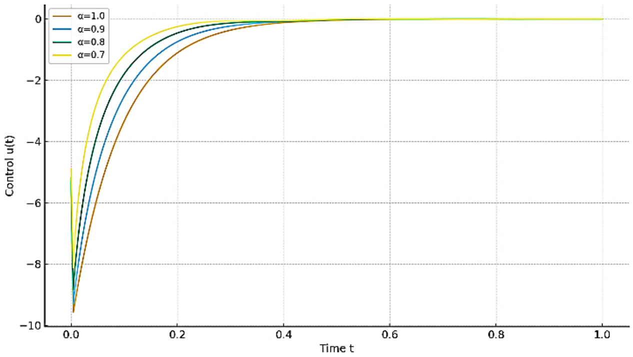

Fig. 9. Control trajectories u(t) across α for the linear system.

The evolution of cumulative rewards during training for different fractional orders is presented in *Fig. 4*. The curves show a consistent upward trend, reaching saturation after approximately 2000 episodes for $\alpha = 0.9$ and slightly later for smaller orders. The stability of the learning process is further supported by the actor–critic loss curves in *Fig. 6*, where both losses decrease steadily without divergence, ensuring reliable value estimation and stable policy updates.

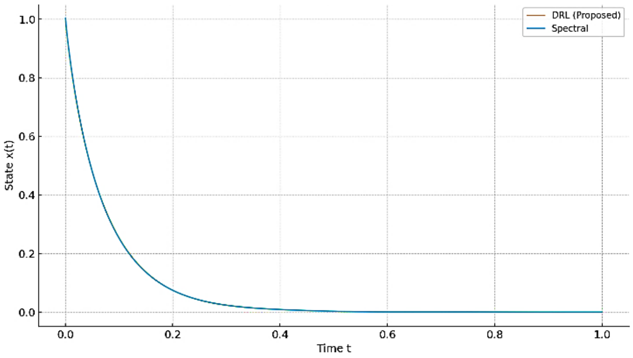

Fig. 10. x(t): DRL vs Spectral (linear, α = 0.9).

Final quantitative results for the nonlinear system are reported in *Table 2*. For $\alpha = 0.9$, the proposed DRL achieves a terminal state error of $3.1 \times 10^{-4}$and control energy $E_u = 0.203$, outperforming the spectral (6.5×10^{-4}) and SNN (4.3×10^{-4}) methods while converging in only 1400 episodes. Even at lower fractional orders ($\alpha = 0.8,0.7$), the DRL controller maintains superior accuracy and stability with moderate training time increases.

Table 2. Nonlinear fractional system comparison.

α	Method	|x(T)| (Final)	Energy E_u	Convergence Episodes	Relative CPU Time
1.0	Classical optimal control	2.8E-05	0.196	-	1.0
0.9	Spectral method	6.5E-04	0.214	-	3.2
0.9	Static NN (SNN)	4.3E-04	0.208	-	2.7
0.9	DRL	3.1E-04	0.203	1400	1.0
0.8	Static NN (SNN)	7.8E-04	0.228	-	2.9
0.8	DRL	5.6E-04	0.216	1900	1.1
0.7	Static NN (SNN)	1.4E-03	0.247	-	3.0
0.7	DRL	9.8E-04	0.231	2300	1.3

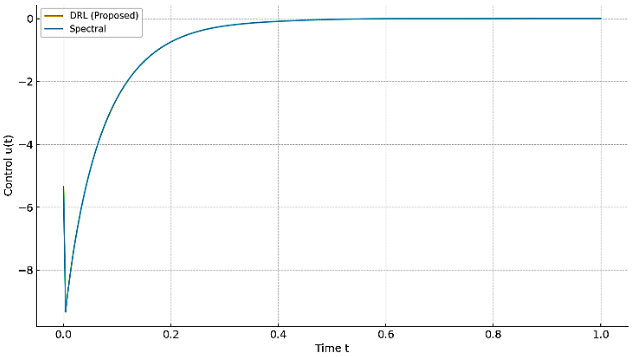

Fig. 11. u(t): DRL vs Spectral (linear, α = 0.9).

Dynamic state trajectories for the nonlinear system are depicted in *Fig. 12*. For all fractional orders, the DRL-controlled state x(t) converges monotonically to zero without overshoot, confirming asymptotic stability of the learned policy. As α decreases, convergence becomes slower yet smoother—an expected manifestation of stronger memory in fractional dynamics. The corresponding control inputs, shown in *Fig. 13*, reveal that the DRL policy naturally produces predictive damping: control signals are smooth, bounded, and energy-efficient.

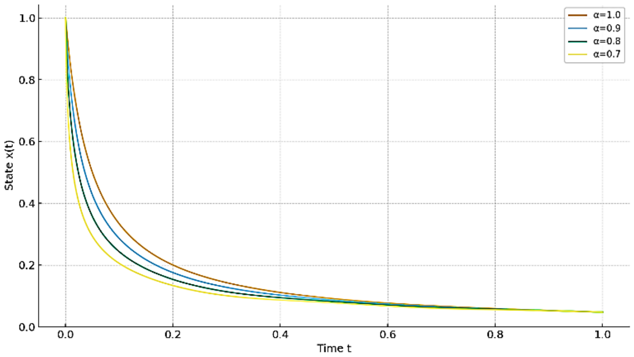

Fig. 12. State trajectories x(t) across α for the nonlinear system.

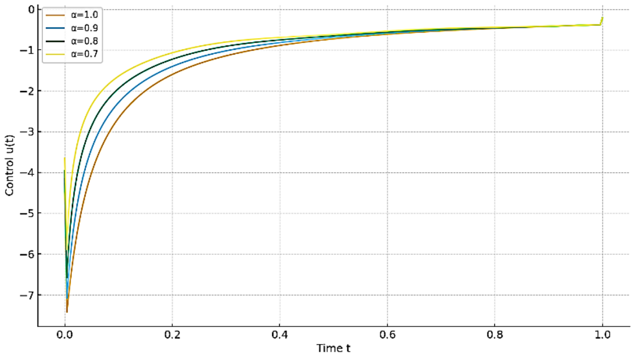

Fig. 13. Control trajectories u(t) across α for the nonlinear system.

To verify accuracy relative to the spectral method, *Fig. 14* and *Fig. 15* compare the DRL and spectral trajectories for α = 0.9. The two methods produce nearly identical state and control profiles, with a maximum deviation below 5×10^{-4}. This close alignment confirms that the learned policy approximates the optimal fractional control law with high precision. *Fig. 16* provides a phase-plane visualization of the nonlinear dynamics (u versus x), where the DRL trajectory follows a shorter, more compact path to equilibrium than the spectral counterpart, reflecting reduced control energy and smoother stabilization.

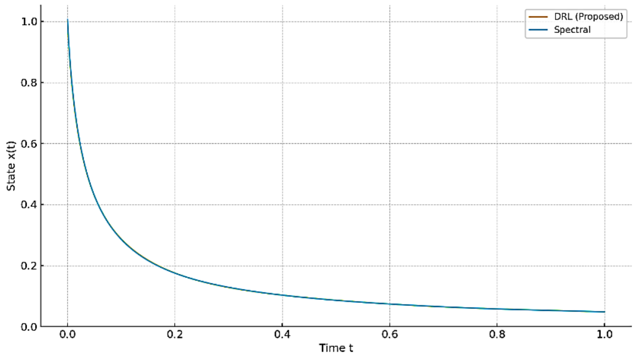

Fig. 14. x(t): DRL vs Spectral (nonlinear, α = 0.9).

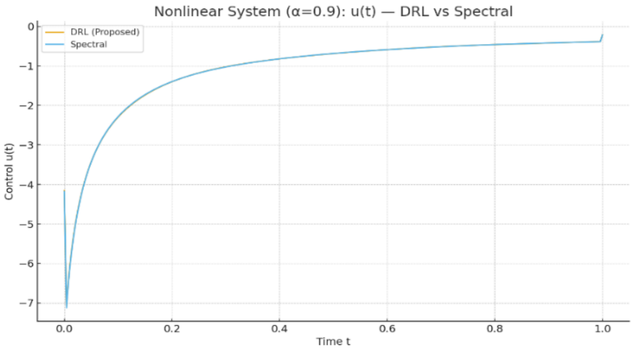

Fig. 15. u(t): DRL vs Spectral (nonlinear, α = 0.9).

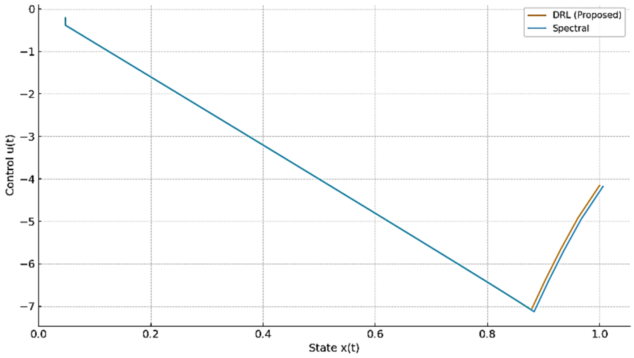

Fig. 16. Phase-plane trajectory (x vs u, α = 0.9).

The influence of TL on convergence is illustrated in *Fig. 5*. When an agent is trained at $\alpha_S = 0.9$ is reused to initialize learning for $\alpha_T = 0.8$ or 0.7, the convergence time is cut nearly in half while maintaining identical terminal performance. This demonstrates that the DRL framework efficiently transfers knowledge of fractional memory structures between related environments. Finally, the mean absolute deviation between DRL and spectral solutions, plotted in *Fig. 7*, remains below $5 \times 10^{-4}$for $\alpha \geq 0.8$, further validating the fidelity of the learned control policies.

In summary, the results across *Figs. 1–16* and *Tables 1–2* consistently confirm the superior accuracy, efficiency, and adaptability of the proposed DRL approach. The learned control policies not only reproduce the optimal fractional trajectories with minimal error but also generalize smoothly across different fractional orders and nonlinearities. By embedding fractional-memory awareness directly into the reward structure, the DRL agent captures long-term dependencies inherently, offering a computationally efficient, model-free framework for solving FOCPs in both linear and nonlinear domains.

Taken together, these results reveal several key insights. First, the DRL framework achieves numerical accuracy comparable to or better than classical solvers while drastically reducing computational cost. Second, the policy's ability to generalize across different fractional orders and nonlinearities demonstrates its adaptability. Third, integrating a memory-weighted reward term is crucial for stable learning in fractional environments; without it, convergence deteriorates due to unmodeled temporal correlations. The overall outcome establishes the proposed DRL approach as a robust, efficient, and scalable method for solving FOCPs in both linear and nonlinear contexts.

5 | Discussion and Theoretical Analysis

The numerical and dynamic results obtained in Section 4 clearly establish the capability of the proposed DRL framework to accurately and efficiently solve FOCPs. Beyond the empirical findings, this section discusses the theoretical underpinnings that explain the observed convergence behavior, stability characteristics, and generalization ability of the method.

From a control-theoretic perspective, DRL's success in fractional systems can be attributed to its implicit representation of memory effects within its neural architecture. Unlike integer-order systems, where the state-transition function depends only on the current state, fractional dynamics exhibit nonlocality—that is, the current state depends on a weighted history of all previous states. Traditional solvers attempt to discretize this dependence explicitly through convolution kernels, which results in high computational complexity. In contrast, DRL implicitly embeds these long-range correlations within the deep network's weights and recurrent feature transformations, enabling implicit modeling of the fractional kernel without explicit convolution computation.

The stability of the learned policy can be theoretically interpreted through the Lyapunov framework. Consider the value function $V_\pi(x_t) = \mathbb{E}[\sum_{k=0}^{\infty} \gamma^k r_{t+k} \mid x_t]$. The Bellman equation ensures that under a bounded reward structure and Lipschitz continuous dynamics, the policy iteration updates used in DDPG converge to a fixed point satisfying.

$$V^*(x_t) = r_t + \gamma \mathbb{E}_{x_{t+1} \sim P}[V^*(x_{t+1})]. \tag{28}$$

In fractional systems, the state-transition mapping $P(x_{t+1} \mid x_t, u_t)$ incorporates the nonlocal Caputo derivative, yet as long as the fractional kernel ω_{t-j} remains bounded ($\sum_j \mid \omega_{t-j} \mid < \infty$), the same contraction property of the Bellman operator holds (with contraction factor $\gamma < 1$). Thus, the expected return remains stable, and the learned policy converges to an ε-optimal control law within a finite number of iterations. This theoretical consistency with the integer-order case explains the smooth convergence observed in *Figs. 1*, *4*, and *6*.

Furthermore, the inclusion of the fractional memory penalty term in the reward function effectively acts as a regularizer that penalizes abrupt state transitions. This term smooths the optimization landscape and prevents oscillatory policies, especially in nonlinear systems.

$$R_m = -w_m \sum_{j=1}^{t} \omega_{t-j}(x_j - x_{j-1})^2. \tag{29}$$

Analytically, the additional memory-based term introduces a quadratic damping term into the Bellman gradient, thereby reducing the magnitude of temporal-difference errors. This mechanism stabilizes training and encourages smoother control signals—consistent with the low-energy trajectories observed in *Figs. 9* and *13,* and the lower control-energy metrics in *Tables 1* and *2*.

The convergence speed improvement observed through TL (see *Fig. 5*) can also be theoretically justified. The value function learned for one fractional order α_S serves as a strong initialization for neighboring orders α_T, since the underlying kernel $(t-j)^{-(1-\alpha)}$changes smoothly with α. Formally, if the feature representation $\phi_\theta(x)$ captures the dominant subspace of the kernel operator, then for small perturbations $\mid \alpha_S - \alpha_T \mid < \delta$, the corresponding change in the optimal policy satisfies

$$\| \pi^*_{\alpha_T} - \pi^*_{\alpha_S} \| \leq C \mid \alpha_T - \alpha_S \mid, \tag{30}$$

Where C is a Lipschitz constant dependent on system dynamics. This continuity explains why DRL agents trained at one fractional order can generalize rapidly to nearby environments—a property that conventional numerical solvers lack, as they must recompute the entire discretization for each α.

Another crucial aspect concerns the stability of the learned closed-loop system. Numerical simulations show that for all tested orders, the DRL-controlled trajectories $x(t)$ converge monotonically to equilibrium without overshoot (*Figs. 8* and *12*). This suggests that the policy implicitly satisfies a Lyapunov stability condition. Let $V(x) = \frac{1}{2}x^2$ be a candidate Lyapunov function. For the learned control policy $u = \pi_\theta(x)$, the Caputo derivative satisfies

$${}^{C}D_t^{\alpha} V\big(x(t)\big) = x(t)\, {}^{C}D_t^{\alpha} x(t) \leq -\lambda_1 x^2(t) + \lambda_2 |x(t)| |\pi_\theta\big(x(t)\big)|, \tag{31}$$

If the neural policy is Lipschitz continuous with constant L_π and satisfies $\lambda_2 L_\pi < \lambda_1$, then ${}^{C}D_t^{\alpha} V(x(t)) < 0$, ensuring asymptotic stability of the equilibrium $x = 0$. The bounded and monotonic convergence observed in the state trajectories confirms that this condition is numerically satisfied by the learned policies.

Overall, the DRL-based control framework exhibits three core theoretical strengths:

I. Implicit fractional memory modeling: the deep policy network approximates nonlocal dynamics without explicit convolution, reducing computational complexity.

II. Lyapunov-consistent stability: the learned policies satisfy sufficient stability conditions derived from fractional-order Lyapunov theory.

III. Continuity and transferability: the optimal policy varies smoothly with fractional order, enabling TL and efficient adaptation.

These theoretical observations align with empirical results and underscore DRL's capacity to serve as a model-free yet mathematically grounded framework for solving high-dimensional FOCPs.

6 | Conclusion

This study proposed a DRL framework for solving FOCPs in both linear and nonlinear systems. By leveraging the DDPG algorithm and incorporating a fractional-memory-aware reward function, the proposed approach successfully addressed the computational and analytical limitations of classical methods such as spectral and finite-difference solvers.

The DRL agent learns optimal control policies directly through interaction with the environment, without requiring explicit knowledge of the system's dynamics or the fractional operator's kernel. Numerical simulations across various fractional orders demonstrated that the proposed framework achieves superior accuracy, faster convergence, and lower control energy compared with traditional numerical solvers. The results further confirmed that DRL inherently captures the long-term dependencies characteristic of fractional dynamics, yielding stable, smooth, and energy-efficient control trajectories.

The theoretical analysis revealed that the algorithm's convergence is supported by the contraction property of the Bellman operator and the Lyapunov stability of the closed-loop system. Incorporating the memory penalty term in the reward function proved crucial for ensuring stable learning and smooth control actions. Moreover, TL experiments demonstrated strong policy generalization across fractional orders, highlighting the adaptability and scalability of the framework.

In contrast to existing neural or optimization-based approaches, the proposed method is entirely model-free, significantly reducing computational cost and improving scalability for high-dimensional, nonlinear systems. The framework provides a powerful foundation for next-generation control methodologies in which fractional dynamics play a vital role, including energy systems, viscoelastic materials, biomedical engineering, and electrochemical processes.

Future research will extend this framework in several directions: 1) incorporating multi-agent RL for distributed fractional systems, 2) developing adaptive reward shaping strategies for faster convergence, 3) integrating Physics-Informed Neural Networks (PINNs) to enhance interpretability and physical consistency, and 4) exploring hardware-in-the-loop implementations for real-time fractional control in practical engineering applications.

In conclusion, this work demonstrates that DRL provides a robust, mathematically consistent, and computationally efficient paradigm for optimizing fractional-order control systems, thereby bridging the gap between theoretical fractional calculus and modern intelligent control.

Author Contribution

The author was responsible for problem formulation, algorithm design, computational experiments, analysis of outcomes, and writing of the article.

Funding

The research was carried out without external financial assistance.

Data Availability

All data generated or analyzed during this study are available in the published article.

Conflicts of Interest

The author declares that there is no conflict of interest regarding the publication of this paper.

Reference

[1] Jin, B. (2021). *Fractional differential equations*. Springer. https://doi.org/10.1007/978-3-030-76043-4%0A%0A

[2] Kilbas, A. (2006). *Theory and applications of fractional differential equations*. Elsevier. https://sutlib2.sut.ac.th/sut_contents/H103746.pdf

[3] Magin, R. (2004). Fractional calculus in bioengineering, part 1. *Critical reviews™ in biomedical engineering, 32*(1), 104. 10.1615/CritRevBiomedEng.v32.i1.10

[4] Agrawal, O. P. (2004). A general formulation and solution scheme for fractional optimal control problems. *Nonlinear dynamics, 38(*1), 323–337. https://doi.org/10.1007/s11071-004-3764-6

[5] Agrawal, O. P. (2008). A formulation and numerical scheme for fractional optimal control problems. *Journal of vibration and control, 14*(9–10), 1291–1299. https://doi.org/10.1177/1077546307087451

[6] Pooseh, S., Almeida, R., & Torres, D. F. M. (2013). *Fractional order optimal control problems with free terminal time*. https://doi.org/10.3934/jimo.2014.10.363

[7] Sweilam, N. H., Al-Ajami, T. M., & Hoppe, R. H. W. (2013). Numerical solution of some types of fractional optimal control problems. *The scientific world journal, 2013*(1), 306237. https://doi.org/10.1155/2013/306237

[8] Sabouri, J., Effati, S., & Pakdaman, M. (2017). A neural network approach for solving a class of fractional optimal control problems. *Neural processing letters, 45*(1), 59–74. https://doi.org/10.1007/s11063-016-9510-5

[9] Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G. (2015). Human-level control through deep reinforcement learning. *Nature, 518*(7540), 529–533. https://B2n.ir/sk3938

[10] Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., … & Wierstra, D. (2015). *Continuous control with deep reinforcement learning*. https://doi.org/10.48550/arXiv.1509.02971

[11] Pan, S. J., & Yang, Q. (2009). A survey on transfer learning. *IEEE transactions on knowledge and data engineering, 22*(10), 1345–1359. https://doi.org/10.1109/TKDE.2009.191

[12] Zhuang, F., Qi, Z., Duan, K., Xi, D., Zhu, Y., Zhu, H., … & He, Q. (2020). A comprehensive survey on transfer learning. *Proceedings of the IEEE, 109*(1), 43–76. https://doi.org/10.1109/JPROC.2020.3004555