



---

*Research article*

## Probability density prediction for linear systems under fractional Gaussian noise excitation using physics-informed neural networks

Baolan Li<sup>1</sup>, Shaojuan Ma<sup>1,2,\*</sup>, Hufei Li<sup>3</sup> and Hui Xiao<sup>1</sup>

<sup>1</sup> School of Mathematics and Information Science, North Minzu University, Yinchuan 750021, China

<sup>2</sup> Ningxia Key Laboratory of Intelligent Information and Big Data Processing, North Minzu University, Yinchuan 750021, China

<sup>3</sup> Shanghai Institute of Applied Mathematics and Mechanics, Shanghai Key Laboratory of Mechanics in Energy Engineering, Shanghai Frontier Science Center of Mechanoinformatics, School of Mechanics and Engineering Science, Shanghai University, Shanghai 200072, China

\* **Correspondence:** Email: [sjma@nmu.edu.cn](mailto:sjma@nmu.edu.cn).

**Abstract:** This study centered on employing the physics-informed neural networks (PINNs) approach for resolving the time-dependent Fokker-Planck-Kolmogorov (FPK) equation for the first time, culminating in the derivation of the transient probability density function. First, we derived the FPK equation for a dynamical system driven by fractional Gaussian noise (FGN). Second, a deep learning method based on PINNs was introduced for resolving the corresponding time-dependent FPK equation. Finally, two examples under two different excitation conditions were discussed to determine the effectiveness and feasibility of the PINNs algorithm. The results show that the PINNs algorithm can get the transient solution of the system under additive and multiplicative FGN. Concurrently, the Monte Carlo approach was utilized to evaluate the precision and computational efficiency of the PINNs algorithm. We found that the different comparison results are in good consistency, which proves that the PINNs algorithm is not only efficient, but also effective and interpretable.

**Keywords:** Fokker-Planck-Kolmogorov equation; physics-informed neural networks algorithm; dynamical system; fractional Gaussian noise; Monte Carlo approach

---

### 1. Introduction

The Fokker-Planck-Kolmogorov (FPK) equations are partial differential equations, which describe the change of the probability density function (PDF) of stochastic processes with time. In physics, the FPK equation is commonly utilized to examine the temporal evolution of the distribution of particle positions or velocities under the influence of stochastic forces within a potential energy field. In

practical application, the FPK equation can be used to describe various stochastic dynamical systems, including those across disciplines like physics, chemistry, biology, and economics.

Normally, through analyzing the transient or steady-state PDF of system response, the instantaneous and aggregate impacts of stochastic excitation upon the system can be quantified, which are described by FPK equations. Hence, resolving FPK equations is very important for interpreting the behavior of dynamical systems. Relatively speaking, an analytical solution to the FPK equation is achievable under certain restrictive conditions, whereas under normal circumstances, the analytical solution for the FPK equation is nearly unattainable [1]. Therefore, approximate and numerical methods, such as the stochastic average method [2], finite element method [3], Monte Carlo method [4], finite difference method [5], path integration method [6], Galerkin method [7], and variational method [8] are usually used to solve the FPK equation. These methods mainly divide the computing domain into grid point sets, and find approximate solutions at these points. Nevertheless, these solutions are only provided at grid points, and assessing any location beyond grid points necessitates the use of interpolation or alternative reconstruction strategies. For low-dimensional systems, analytical approximations like the exponential polynomial closure method [9] have shown their effectiveness and efficiency. When dealing with high-dimensional systems, the state-space-split method [10] has shown its effectiveness and efficiency.

In contrast to conventional techniques, machine learning exhibits superior potential when addressing high-dimensional differential equations. Li et al. [11] studied the theoretical analysis of stochastic dynamics and introduced an artificial neural network (ANN) method for resolving stochastic partial differential equations (SPDE) via the FPK equation or adjoint FPK equation, demonstrating enhanced robustness and precision. Xiao et al. [12] proposed a neural network technique for analyzing stochastic vibrations in large, nonlinear systems with Gaussian white noise. Their approach simplifies the steady-state Fokker-Planck equation from high to low dimensions, showing that the simplified coefficients are conditional means of the original coefficients. Raissi et al. [13] put forward a deep learning architecture of PINNs, which combines a physical model and data, making it suitable for continuous and discrete time models. Later, Rao et al. [14] applied a mixed variable output PINN to simulate elastic dynamics, improve network performance, verify various problems, and show the application prospect of computational mechanics. Chen et al. [15] proposed a PINN to discover control equations from sparse data, which integrates the advantages of deep learning and effectively identifies system behavior, which is suitable for data scarcity scenarios. Therefore, the neural network based on physics has become a research hotspot. As deep learning advances, the application of neural networks has extended into the domain of stochastic dynamics, especially in solving FPK equations [16]. Wang et al. [17, 18] investigated the time-dependent PDF of dynamical systems under Gaussian white noise influence, employing a gaussian radial basis function neural network (GRBFNN). The adaptive gaussian mixture model (AGMM) machine learning method was proposed by Sun and his associates, which was designed to tackle the general FP equation, in comparison with the previous numerical discretization method. It created a seamless collection of data and mathematical models [19]. Zhang and Yuen [20] applied the deep learning method based on physical guidance to address the time-varying FPK equation, incorporating physical modeling as a guiding constraint to direct the deep neural network in learning the solutions to the equation. Xu et al. [21] introduced a technique for solving the FPK equation using a deep neural network that eliminates the need for interpolation and coordinate transformation, distinct from

conventional numerical techniques. To decrease the sensitivity of the FPK equation to boundary conditions, Li [22] presented a new data-driven method to solve the FPK equation in any local region. Inspired by the result [22], Zhai et al. [23] used an ANN for solving the FPK equation.

The aforementioned studies primarily concentrate on the Gaussian white noise model. As research into stochastic phenomena in related fields progresses, the limitations of the Gaussian white noise model in describing many stochastic dynamics problems have been gradually revealed. Especially in the financial market, the fluctuation of stock price is often influenced by financial and economic factors with time correlation, and this excitation is more suitable for accurate modeling through FGN. Mandelbrot and Van Ness [24] introduced fractional Brownian motion (FBM) in 1968, outlining its creation. Subsequently, research on dynamical systems driven by FBM has garnered significant scholarly interest. FBM is not classified as a semi-martingale or a Markov process, so the complete theoretical basis of stochastic integration is not suitable for the FBM. Therefore, a new stochastic calculus theory is needed to deal with FBM. Duncan et al. [25] put forward a stochastic calculus theory suitable for FBM with a Hurst index in the  $(1/2, 1)$  interval. Vas'kovskii [26] further extended the theory by providing the FPK equation for a one-dimensional stochastic differential equation with FBM as the driver, for a Hurst index within  $(0, 1)$ . This finding broadens the scope of previous findings for one-dimensional stochastic differential equation excited by FBM, which were limited to a Hurst index in the interval  $(1/3, 1)$  [27, 28]. Xu et al. [29] studied the stochastic averaging principle under FBM under the framework of forward path integration, and proved the convergence for the dynamical system excited by FBM. Pei et al. [30] further studied the stochastic average principle of the fast-slow system influenced by FBM. Deng and Zhu [31] introduced a stochastic averaging technique for quasi-Hamilton systems with single or multiple degrees of freedom under the FGN, which effectively solved the average equation of quasi-Hamilton systems. A generalized hat function was proposed to solve nonlinear stochastic differential equation excited by multiplicative FGN [32]. Therefore, it is effective and necessary to introduce the FGN model into dynamical systems to describe stochasticity with a time correlation and probability distribution characteristics.

In conclusion, it is found that the application of deep learning technology in stochastic dynamics has not been extensively deepened, particularly for addressing FPK equations under the influence of FGN. Current investigations have largely centered on systems stimulated by Gaussian or non-Gaussian white noise, with the FPK equations perturbed by FGN being a less explored area. Therefore, inspired by the discussion above, we introduce FGN into the dynamical system in this paper, derive the FPK equation of the stochastic system, and then propose a deep learning theoretical system based on PINNs to solve the FPK equation. Unlike prior studies focused on Gaussian white noise [12, 17, 18], our work addresses the critical gap in modeling time-correlated FGN excitations, which are prevalent in real-world systems but lack efficient computational tools. The primary contributions and novel aspects of this research are highlighted in the following critical areas:

- 1) In the study of dynamical systems, it is assumed that the stochastic perturbation is Gaussian white noise for the convenience of analysis. In reality, FGN can reflect the time correlation of stochasticity in a dynamical system. Therefore, the FGN with a time correlation is brought into the analysis of stochastic dynamical systems in this paper.
- 2) We include a rigorous derivation of FGN-induced FPK equations via Malliavin-Wick calculus, establishing new theoretical foundations, extending beyond traditional Gaussian white

noise models.

- 3) We provide the first application of PINNs to solve the FPK equation for dynamical systems driven by FGN, addressing a gap in existing research.
- 4) Compared to existing methods, this work provides the first PINNs-based framework for FGN-driven systems, achieving higher computational efficiency (100 times faster than Monte Carlo) while maintaining accuracy ( $< 1\%$  error).

The structure of the remainder of this paper is as follows. Section 2 outlines the fundamental principles of FBM and FGN, including the stochastic integral and Malliavin calculus for FBM. Subsequently, Section 3 derives the Fokker-Planck equation for a dynamical system driven by FGN. Section 4 describes the method of using PINNs to solve the corresponding time-varying FPK equation, and gives the algorithm scheme. In Section 5, we present two stochastic dynamical systems as case studies to illustrate the efficacy and clarity of the PINNs method. Section 6 gives a brief conclusion.

## 2. Preliminaries

FBM is considered an extension of standard Brownian motion within the realm of probability theory. Compared with the typical BM, FBM is characterized in that the increments are not necessarily independent. The purpose of this section is to introduce the definition of FBM, discuss its properties and related knowledge of FGN.

### 2.1. Definition of FBM and its properties

The concept of FBM was coined in 1968 by Mandelbrot and Van Ness [24]. The stochastic integral representation of the process based on standard BM has been provided in literature [24], where the stochastic integral is defined by the Hurst index  $H$ :

$$B^{(H)}(f) = C_H \left\{ \int_{-\infty}^0 [(f-g)^{H-\frac{1}{2}} - (-g)^{H-\frac{1}{2}}] dB(g) + \int_0^f (f-g)^{H-\frac{1}{2}} dB(g) \right\}, \quad (2.1)$$

with  $B(g)$  denoting the standard Brownian motion and  $C_H$  representing the normalization constant.

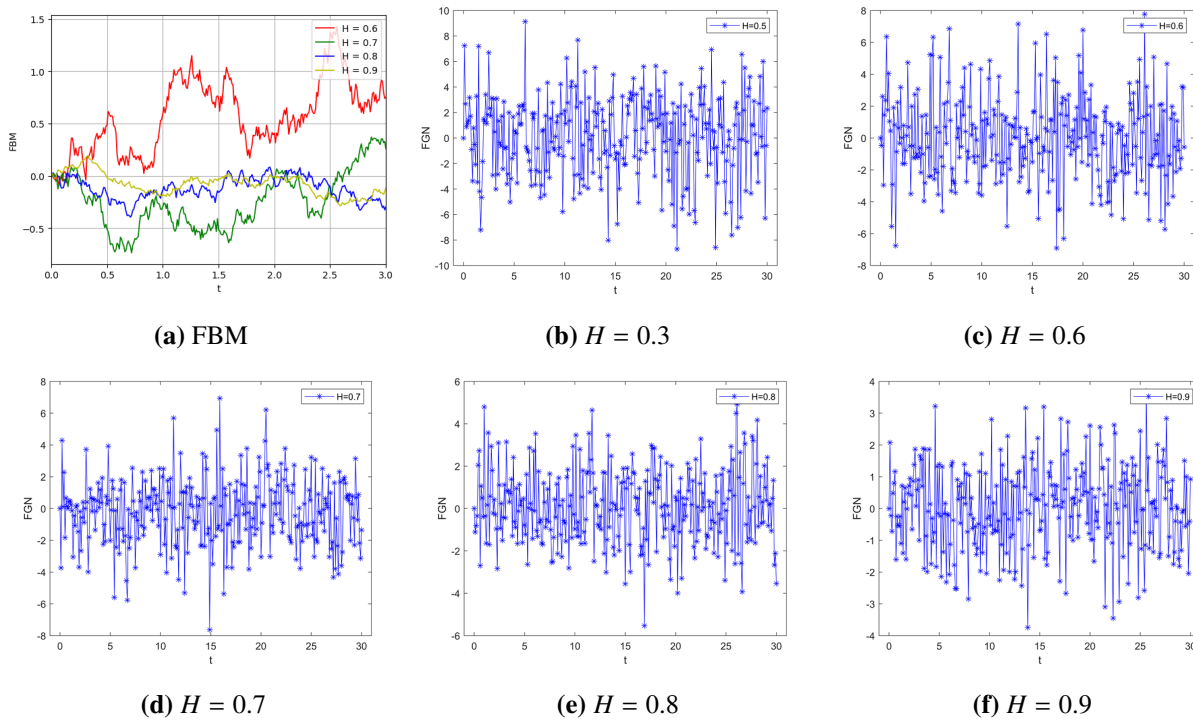
**Definition 2.1.** Given a complete probability space  $(\Omega, \mathcal{F}, \mathbb{P})$ , let the Hurst parameter  $H \in (0, 1)$  be a real number. According to literature [33], its one-dimensional FBM  $\{B^{(H)}(s), s \geq 0\}$  with Hurst index  $H$  is characterized as a mean-zero, continuous Gaussian process with the property of having a covariance function:

- 1)  $E[B^{(H)}(f)] = 0, \forall f \geq 0$ ,
- 2)  $E[B^{(H)}(f)B^{(H)}(g)] = 1/2(f^{2H} + g^{2H} - |f-g|^{2H})$ .

For all  $\alpha \in (0, H)$ , the sample trajectories of the FBM are almost surely  $\alpha$ -order Hölder continuous. The size of the real number  $H$  determines what kind of process the FBM is:

When  $H < 1/2$ , then the increments of the process are negatively correlated and  $\{B^{(H)}(f), f \geq 0\}$  has an anti-persistence. When  $H = 1/2$ , FBM  $\{B^{(H)}(f), f \geq 0\}$  is a standard Brownian motion or

Wiener process. When  $H > 1/2$ , then the increments of the process are positively correlated and  $\{B^{(H)}(f), f \geq 0\}$  has long-range dependence. Figure 1 shows the sample paths of FBM and FGN under different Hurst indices  $H$ .



**Figure 1.** Sample paths of FBM and FGN.

**Properties 2.1.** By Definition 2.1 and literature [34], a standard FBM is obtained, which has the following characteristics:

- 1)  $B^{(H)}(0) = 0$ .
- 2)  $B^{(H)}(f)$  has homogeneous increments, i.e.,  $B^{(H)}(f + g) - B^{(H)}(g)$  has the same law for  $f, g \geq 0$ .
- 3)  $B^{(H)}(f)$  is a Gaussian process and  $E[B^{(H)}(f)^2] = f^{2H}$ ,  $E[dB^{(H)}(f)^2] = (df)^{2H}$ ,  $s \geq 0$ , for all  $H \in (0, 1)$ .
- 4) The sample trajectories of  $B^{(H)}(f)$  are continuous but almost everywhere non-differentiable.
- 5)  $\{B^{(H)}(f), f \geq 0\}$  has self-similarity, i.e.,  $\{B^{(H)}(\beta f), f \geq 0\}$  has the same probability law as  $\{\beta^H B^{(H)}(f), f \geq 0\}$ .
- 6) When  $1/2 < H < 1$ ,  $\{B^{(H)}(f), f \geq 0\}$  has long-range dependence, so that if

$$R(m) = \text{Cov}[B^{(H)}(1), B^{(H)}(m+1) - B^{(H)}(m)],$$

there is  $\sum_{m=1}^{\infty} R(m) = \infty$ .

A unit-FGN can be conceptualized as the derivative of a unit-FBM:

$$B^{(H)}(f) = \int_0^f W^{(H)}(g) dg,$$

or

$$W^{(H)}(f) = \frac{dB^{(H)}(f)}{df} = \frac{B^{(H)}(f + \Delta f) - B^{(H)}(f)}{\Delta f}. \quad (2.2)$$

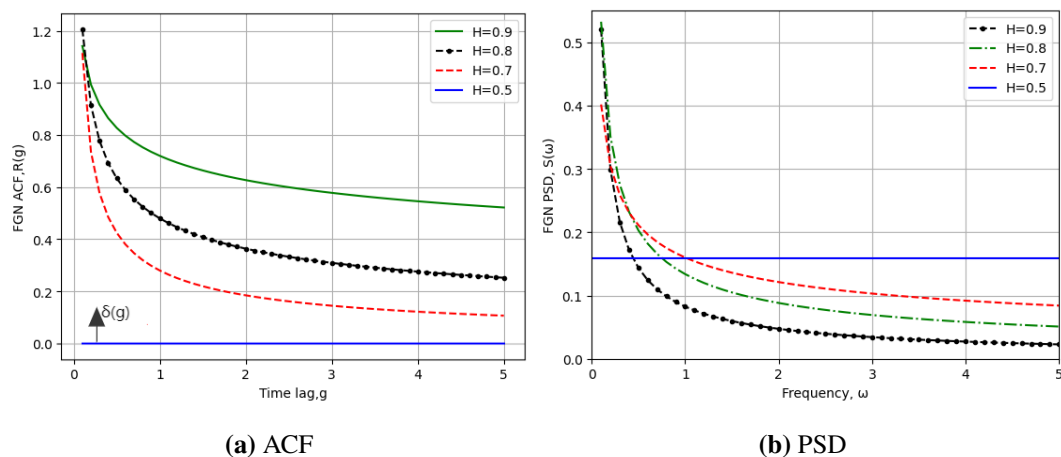
Its autocorrelation function (ACF) is

$$R(g) = E \left[ W^{(H)}(f + g) W^{(H)}(f) \right] = H(H + 1) |g|^{2H-2} + 2H |g|^{2H-1} \delta(g).$$

As the power spectral density (PSD) of FGN holds physical significance exclusively within the range  $1/2 < H < 1$ , the study here is limited to this interval, focusing on the corresponding PSD.

$$S(w) = \frac{1}{2\pi} \int_{-\infty}^{\infty} R(k) e^{-i w k} dk = \frac{H \Gamma(2H) \sin(H\pi)}{\pi} |w|^{1-2H}.$$

Figure 2 shows the numerical results of ACF  $R(g)$  and PSD  $S(w)$  of FGN when  $1/2 < H < 1$ . As can be seen from the figure, when  $H$  tends to  $1/2$ , the PSD of FGN is close to a constant and the ACF is close to the Dirac function  $\delta(g)$ . When  $H$  tends to 1, the PSD is close to the Dirac function  $\delta(g)$  and the ACF is close to a constant. When  $H$  is between  $1/2$  and 1, the ACF has a finite value for a period of time, which indicates that FGN has a long-range correlation. Therefore, when  $1/2 < H < 1$ , FGN is a stationary Gaussian process between Gaussian white noise and Gaussian random variables.



**Figure 2.** The ACF and PSD of FGN.

## 2.2. Fractional stochastic calculus

Given a fixed Hurst index of  $1/2 < H < 1$ , consider  $\mathbb{R}_+ \times \mathbb{R}_+ \rightarrow \mathbb{R}_+$  defined as follows:

$$\phi(s, t) = H(2H - 1) |s - t|^{2H-2}.$$

Let  $g : \mathbb{R}_+ \rightarrow \mathbb{R}$  be a Borel-measurable (deterministic) function. The function  $g$  is an element of the Hilbert space  $L_\phi^2(\mathbb{R}_+)$  provided that

$$\|g\|_\phi^2 := \int_0^\infty \int_0^\infty g(s)g(t)\phi(s,t)dsdt < \infty.$$

The inner product within the Hilbert space  $L_\phi^2(\mathbb{R}_+)$  is represented by  $\langle \cdot, \cdot \rangle_\phi$ .

**Lemma2.1.** *If  $f, g$  belong to  $L_\phi^2(\mathbb{R}_+)$ , then  $\int_0^\infty f(s)dB^{(H)}(s)$ ,  $\int_0^\infty g(t)dB^{(H)}(t)$  are well-defined, zero mean, Gaussian stochastic variables  $\|f\|_\phi^2$ ,  $\|g\|_\phi^2$ , respectively, and*

$$E \left[ \int_0^\infty f(s)dB^{(H)}(s) \int_0^\infty g(t)dB^{(H)}(t) \right] = \int_0^\infty \int_0^\infty f(s)g(t)\phi(s,t)dsdt = \langle f, g \rangle_\phi. \quad (2.3)$$

Lemma 2.1 is verified in literature [35]. Define a mapping  $L_\phi^2(\mathbb{R}_+)$ :

$$(\Phi g)(t) = \int_0^\infty \phi(s,t)g(s)ds, \forall g \in L_\phi^2(\mathbb{R}_+).$$

**Definition2.3.** (Malliavin derivative) *Let  $g$  belong to  $L_\phi^2(\mathbb{R}_+)$ . The  $\phi$ -derivative of a stochastic variable  $F \in L^p(\mathbb{P}^H)$  in the direction of  $\Phi g$  is defined as*

$$D_{\Phi g}F(\omega) = \lim_{\delta \rightarrow 0} \frac{1}{\delta} \left\{ F \left( \omega + \delta \int_0^\infty (\Phi g)(u)du \right) - F(\omega) \right\},$$

*if the limit exists in  $L^p(\Omega, \mathcal{F}, \mathbb{P})$ . Furthermore, if there is a process  $(D^\phi F_s, s \geq 0)$  such that*

$$D_{\Phi g}F = \int_0^\infty D_s^\phi F(s)g(s)ds, \quad a.s.$$

*for all  $g \in L_\phi^2(\mathbb{R}_+)$ , then  $F$  is said to be  $\phi$ -differentiable.*

**Definition2.4.** (The Wick product) *Let  $K(\omega) = \sum_\alpha c_\alpha \mathcal{H}_\alpha(\omega)$  and  $L(\omega) = \sum_\beta d_\beta \mathcal{H}_\beta(\omega)$  be two members of  $L_\phi^2(\mathbb{R}_+)$ . Then we define the Wick product  $K \diamond L$  of  $K$  and  $L$  by*

$$(K \diamond L)(\omega) = \sum_{\alpha\beta} c_\alpha d_\beta \mathcal{H}_{\alpha+\beta}(\omega) = \sum_\xi \left( \sum_{\alpha+\beta=\xi} c_\alpha d_\beta \right) \mathcal{H}_\xi(\omega).$$

### 2.3. Itô formula

**Lemma2.2.** *Let  $\{F(s), s \in [0, T]\}$  be a stochastic process in  $\mathcal{L}_\phi(0, T)$  and denote*

$$\eta_t = \mu + \int_0^t G(s)ds + \int_0^t F(s)dB^{(H)}(s), \mu \in \mathbb{R}.$$

*Consider  $f : \mathbb{R}_+ \times \mathbb{R} \rightarrow \mathbb{R}$  a function that is first continuously differentiable with respect to its first argument and second continuously differentiable with respect to its second argument. Suppose that these derivatives are bounded. Moreover, it is assumed that  $E \left[ \int_0^T |F(s)D_s^\phi \eta_s| ds \right] < \infty$  and  $\left\{ \frac{\partial f}{\partial x}(s, \eta_s) F(s), s \in [0, T] \right\}$  is in  $\mathcal{L}_\phi(0, T)$ . Then for  $0 \leq t \leq T$ , the fractional Itô formula is obtained:*

$$f(t, \eta_t) = f(0, 0) + \int_0^t \frac{\partial f}{\partial s}(s, \eta_s)ds + \int_0^t \frac{\partial f}{\partial x}(s, \eta_s)G(s)ds + \int_0^t \frac{\partial f}{\partial x}(s, \eta_s)F(s)dB^{(H)}(s)$$

$$+ \int_0^t \frac{\partial^2 f}{\partial x^2}(s, \eta_s) F(s) \int_0^s \phi(s, v) F(v) dv ds$$

almost surely, or formally,

$$\begin{aligned} f(t, \eta_t) &= \frac{\partial f}{\partial t}(t, \eta_t) ds + \frac{\partial f}{\partial t}(t, \eta_t) G(t) dt + \frac{\partial f}{\partial x}(s, \eta_s) F(s) dB^{(H)}(s) \\ &\quad + \frac{\partial^2 f}{\partial x^2}(t, \eta_t) F(t) \int_0^t \phi(t, v) F(v) dv dt, \quad a.s. \end{aligned}$$

where  $\phi(s, v) = H(2H - 1) |s - t|^{2H-2}$ .

**Lemma 2.3.** Let  $\{F^i(s), i = 1, \dots, n, s \in [0, T]\}$  satisfy the conditions of Lemma 2.2 for  $F$ ,  $t \in [0, T]$ . Let

$$\mu_t^j = \int_0^t F^j(s) dB^{(H)}(s), \quad j = 1, 2, \dots, n$$

for  $j = 1, 2, \dots, n$ , and let  $\left\{ \frac{\partial f}{\partial x_j}(s, \mu_s) F^j(s), s \in [0, T] \right\}$  be in  $\mathcal{L}(0, T)$ . Assuming  $f : \mathbb{R}_+ \times \mathbb{R}^n \rightarrow \mathbb{R}$  has bounded second-order derivatives and is twice continuously differentiable, then

$$\begin{aligned} f(t, \mu_t^1, \dots, \mu_t^n) &= f(0, 0, \dots, 0) + \int_0^t \frac{\partial f}{\partial s}(s, \mu_s^1, \dots, \mu_s^n) ds \\ &\quad + \sum_{j=1}^n \int_0^t \frac{\partial f}{\partial x_j}(s, \mu_s^1, \dots, \mu_s^n) F^j(s) dB^{(H)}(s) \\ &\quad + \sum_{j,l=1}^n \int_0^t \frac{\partial^2 f}{\partial x_j \partial x_l}(s, \mu_s^1, \dots, \mu_s^n) F^j(s) D^\phi(s) \mu^l(s) ds. \quad a.s. \end{aligned}$$

Particularly, if  $\{F^i(s), i = 1, \dots, n, s \in [0, T]\} \in L_\phi^2$ , then we have

$$\begin{aligned} f(t, \mu_t^1, \dots, \mu_t^n) &= f(0, 0, \dots, 0) + \int_0^t \frac{\partial f}{\partial s}(s, \mu_s^1, \dots, \mu_s^n) ds \\ &\quad + \sum_{j=1}^n \int_0^t \frac{\partial f}{\partial x_j}(s, \mu_s^1, \dots, \mu_s^n) F^j(s) dB^{(H)}(s) \\ &\quad + \sum_{j,l=1}^n \int_0^t \frac{\partial^2 f}{\partial x_j \partial x_l}(s, \mu_s^1, \dots, \mu_s^n) F^j(s) \left( \int_0^s F^l(v) \phi(s, v) dv \right) ds. \quad a.s. \end{aligned}$$

### 3. Derivation of the FPK equation driven by FGN

As for stochastic differential equation driven by FBM:

$$dZ(t) = A(t, x)dt + C(t, x)dB^{(H)}(t), \quad (3.1)$$

where  $A(t, x)$  is a drift function and  $C(t, x)$  is a diffusion function.  $A(t, x)$  and  $C(t, x)$  must be linear and one-dimensional when the stochastic differential equation is one-dimensional [36].  $B^{(H)}(t)$  is a standard FBM with Hurst index  $H$ . Furthermore, we assume that  $A(t, x)$  and  $C(t, x) \neq 0$  have the indicated



derivatives. Then, utilizing Lemma 2.2, which is the fractional Itô formula, the scalar function  $h(x)$  is expressed as follows:

$$dh(x) = \left( \frac{dh(x)}{dx} A(t, x) dt + \frac{d^2 h(x)}{dx^2} C(t, x) \int_0^t C(t, x) \phi(s, t) ds \right) dt + \frac{dh(x)}{dx} C(t, x) dB^{(H)}(t), \quad (3.2)$$

where the function  $\phi(s, t)$  is defined by

$$\phi(s, t) = H(2H - 1) |s - t|^{2H-2}.$$

Taking the expectation of Eq (3.2), we can obtain

$$E [dh(x)] = E \left[ \frac{dh(x)}{dx} A(t, x) dt + \frac{d^2 h(x)}{dx^2} C(t, x) \int_0^t C(t, x) \phi(s, t) ds \right] dt + E \left[ \frac{dh(x)}{dx} C(t, x) dB^{(H)}(t) \right]. \quad (3.3)$$

Therefore, we get

$$E [dh(x)] = E \left[ \frac{dh(x)}{dx} A(t, x) dt + \frac{d^2 h(x)}{dx^2} C(t, x) \int_0^t C(t, x) \phi(s, t) ds \right] dt. \quad (3.4)$$

Suppose  $p(t, x)$  represents the PDF of a particle's position at time  $t$  and location  $x$ , and we have

$$E [h(x)] = \int_{-\infty}^{\infty} h(x) p(t, x) dx, \quad (3.5)$$

which means that

$$E \left[ \frac{dh(x)}{dt} \right] = \int_{-\infty}^{\infty} h(x) \frac{\partial p(t, x)}{\partial t} dx. \quad (3.6)$$

Hence, we further obtain that

$$\int_{-\infty}^{\infty} h(x) \frac{\partial p(t, x)}{\partial t} dx = \int_{-\infty}^{\infty} \left( \frac{dh(x)}{dx} A(t, x) dt + \frac{d^2 h(x)}{dx^2} C(t, x) \int_0^t C(t, x) \phi(s, t) ds \right) p(t, x) dx. \quad (3.7)$$

After partial integration, we find that

$$\int_{-\infty}^{\infty} \frac{dh(x)}{dx} A(t, x) p(t, x) dx = A(t, x) p(t, x) h(x) \Big|_{-\infty}^{\infty} - \int_{-\infty}^{\infty} h(x) \frac{\partial (A(t, x) p(t, x))}{\partial x} dx. \quad (3.8)$$

The first term on the right side of Eq (3.7) must vanish since  $p(x, t) = 0$  at  $x = \pm\infty$ . Then

$$\int_{-\infty}^{\infty} \frac{dh(x)}{dx} A(t, x) p(t, x) dx = - \int_{-\infty}^{\infty} h(x) \frac{\partial (A(t, x) p(t, x))}{\partial x} dx. \quad (3.9)$$

Similarly, we handle the second integral in Eq (3.7) to obtain

$$\int_{-\infty}^{\infty} \frac{d^2 h(x)}{dx^2} C(t, x) \int_0^t C(t, x) \phi(s, t) p(t, x) dx = \int_{-\infty}^{\infty} h(x) \frac{\partial^2 (C(t, x) \int_0^t C(t, x) \phi(s, t) p(t, x) ds)}{\partial x^2} dx. \quad (3.10)$$

By reinserting Eqs (3.9) and (3.10) into Eq (3.7), we derive the subsequent equation:

$$\int_{-\infty}^{\infty} h(x) \left( \frac{\partial p(t, x)}{\partial t} + \frac{\partial [A(t, x) p(t, x)]}{\partial x} - \frac{\partial^2 (C(t, x) \int_0^t C(t, x) \phi(s, t) p(t, x) ds)}{\partial x^2} \right) dx = 0. \quad (3.11)$$

Consequently, we can infer

$$\frac{\partial p(t, x)}{\partial t} + \frac{\partial [A(t, x)p(t, x)]}{\partial x} - \frac{\partial^2 \left( C(t, x) \int_0^t C(t, x) \phi(s, t) p(t, x) ds \right)}{\partial x^2} = 0. \quad (3.12)$$

Due to

$$\int_0^t \phi(s, t) ds = \int_0^t H(2H - 1) |s - t|^{2H-2} ds = Ht^{2H-1}, \quad (3.13)$$

substituting Eq (3.13) into Eq (3.12) and merging the diffusion terms, the corresponding FPK equation of Eq (3.1) under FGN excitation is derived as follows:

$$\frac{\partial p(t, x)}{\partial t} + \frac{\partial [A(t, x)p(t, x)]}{\partial x} - Ht^{2H-1} \frac{\partial^2 \left( C(t, x) \int_0^t C(t, x) p(t, x) ds \right)}{\partial x^2} = 0, \quad (3.14)$$

where  $1/2 < H < 1$ , and  $p(t, x)$  is the PDF evolving with time.

In the interest of convenient analysis, of course,  $x \in \Omega$ ,  $t \in [0, T]$ , we can rewrite Eq (3.14) as

$$\frac{\partial p(t, x)}{\partial t} + \mathcal{N} [p(t, x), p_x(t, x), p_{xx}(t, x)] = 0, \quad (3.15)$$

where  $p(t, x)$  is defined on a domain  $\Omega \times [0, T]$ , and  $\mathcal{N} [p(t, x), \theta]$  is the FPK differential operator.

This is known as the FPK equation driven by the stochastic differential Eq (3.1) because the diffusion term in Eq (3.1) is an FBM. Interestingly, when  $H = 1/2$ , Eq (3.14) simplifies to the classical FPK equation. The FPK Eq (3.14) is a parabolic partial differential equation in two dimensions, with initial and boundary conditions given as

$$p(0, x) = \mathcal{I}(x), \quad x \in \Omega, \quad (3.16)$$

$$p(t, x) = 0, \quad x \in \varepsilon\Omega, \quad t \in [0, T], \quad (3.17)$$

where  $\varepsilon\Omega$  denotes the boundary of  $\Omega$ , and  $\mathcal{I}(x)$  represents the initial Dirac delta distribution.

The normalization condition for the PDF must also be met:

$$\int p(t, x) dx = 1. \quad (3.18)$$

#### 4. Solving the FPK equation based on PINNs

This section presents a PINN-based method for solving the time-dependent FPK equation. PINNs integrate physical laws into neural network training, enhancing prediction accuracy and ensuring physical consistency. By embedding energy conservation, momentum principles, and other physical constraints into the loss function, PINNs achieve a deep fusion of data-driven learning and physical knowledge. This approach enables accurate modeling of complex phenomena, even with incomplete system understanding, outperforming traditional data-driven methods in generalization, especially under data scarcity or noise. PINNs combine theoretical cognition (physical laws as learning objectives) and empirical measurement (numerical/experimental data), improving interpretability and approximation accuracy. The FPK equation's physical information is categorized into: regular information (implicit, Eq (3.16)) and numerical information (explicit, Eqs (3.17) and (3.18)). Neural

networks are trained via sampled data to fit both information types, approximating FPK solutions. Training balance is controlled by adjusting sampling scales and loss weights for each information type. The final network approximates all FPK solutions (Figure 3).

Based on Lu et al.'s [37] general PDE-solving steps, Figure 4 illustrates the FPK equation-solving process. The FPK equation's physical information is split into: 1) internal conditions of Eq (3.15) and 2) numerical data for initial and boundary conditions of Eqs (3.16) and (3.17). A neural network within parameter space  $\Xi$  is constructed to encapsulate the solution. Network fitting loss combines neural network output with these two information types. Training samples are balanced, and comprehensive information loss is constructed based on training intensity balance. Through iterative training, the network approximates the equation's physical information. The optimal solution  $p(t|\theta, x), \theta \in \Xi$  minimizes this loss. Integrating regularization and numerical information enhances network interpretability, reflecting the equation's information satisfaction and the network's capacity to estimate physical attributes.

Generally speaking, the process of solving the FPK equation by the PINNs method can be summarized as the following four stages:

1) Neural network architecture design. A neural network model is constructed with two independent variables  $x$  and  $t$ . The output layer of the model is designed to predict the equation solution  $p(t, x|\theta)$ , where  $\theta \in \Xi$  represents the set of parameters in the parameter space  $\Xi$ . The neural network aims to solve the characterization equation  $p(t, x|\theta)$  in the parameter space  $\Xi$ . The neural network's structure is detailed in Table 1, and the Adam optimizer is employed for model training. The selection of the tanh activation function and two hidden layers with 50 neurons each was based on extensive empirical testing and the following considerations:

(a) Activation function:

- i. Tanh was chosen over ReLU due to its smooth derivatives, which are crucial for accurately computing higher-order PDE terms in the FPK equation through automatic differentiation.
- ii. Compared to sigmoid, tanh provides better gradient flow in deep networks by centering the output around zero.
- iii. The softplus output activation ensures non-negative probability density values while maintaining differentiability.

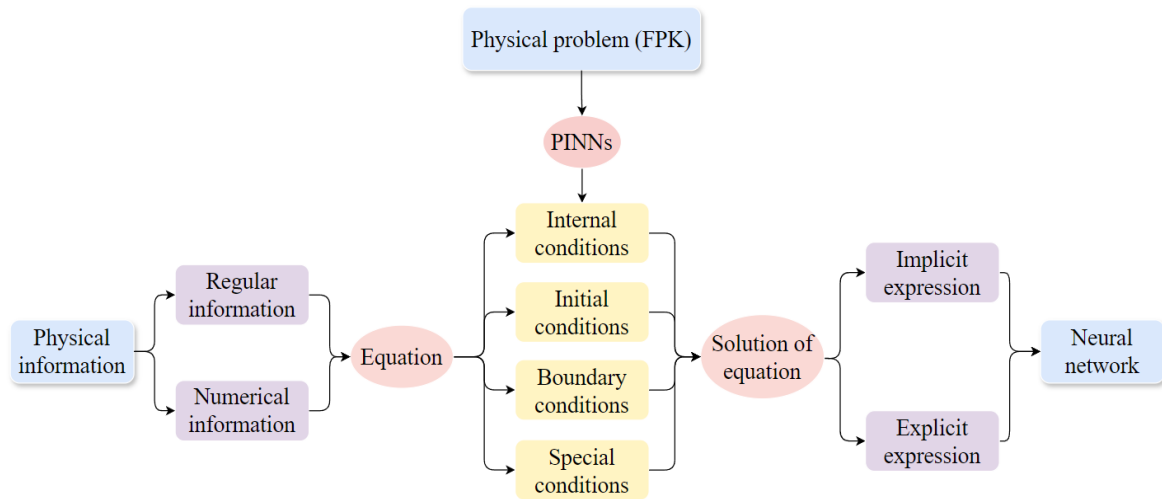
(b) Network depth and width [21]:

- i. Two hidden layers were found to provide sufficient capacity to approximate the solution while avoiding overfitting, as evidenced by ablation studies (Section 5.3).
- ii. 50 neurons per layer achieved optimal balance between computational efficiency (0.4s training time) and accuracy ( $< 1\%$  error), with diminishing returns observed when increasing to 100 neurons.
- iii. Shallower networks (1 hidden layer) underfitted the solution, while deeper networks (3+ layers) showed minimal improvement but required  $2-3 \times$  longer training times.

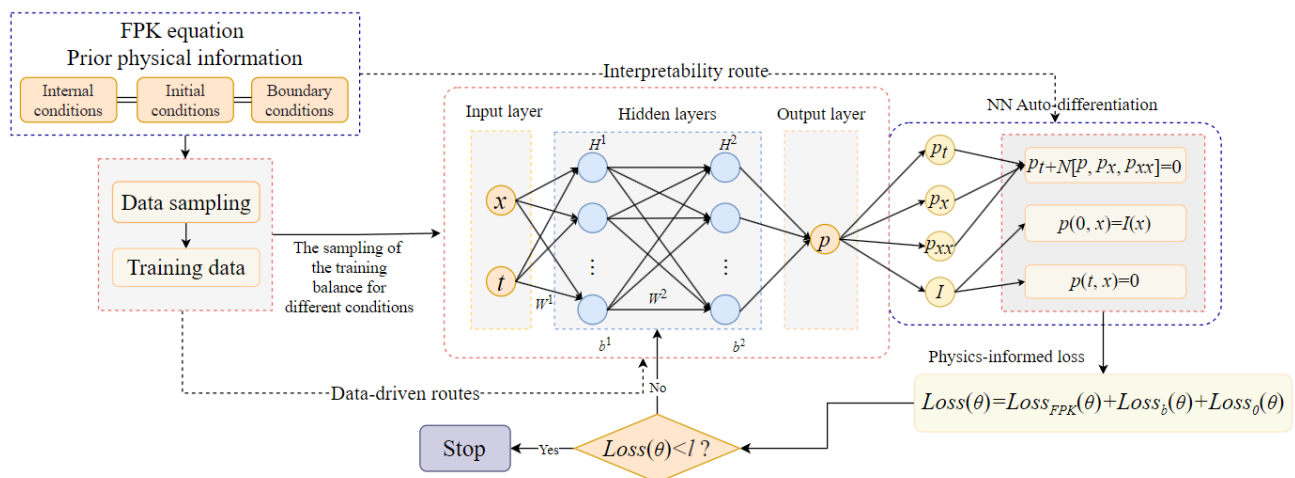
(c) Training stability:

- i. This architecture maintained stable training across different Hurst indices ( $H = 0.5-0.9$ ) and noise intensities ( $\sigma$ ).

- ii. The combination of tanh activations with initialization prevented vanishing or exploding gradients during backpropagation of the physics-informed loss terms.



**Figure 3.** Logical explanation of solving the FPK equation with PINNs.



**Figure 4.** Schematic of PINNs solving the FPK equation.

**Table 1.** Architecture of the neural network.

Layer	Neuron	Activation function
Input layer	$x$	—
	$t$	—
Hidden layer	hidden layer 1	tanh
	hidden layer 2	tanh
Output layer	$\hat{p}(t, x \theta)$	softplus

- 2) Loss function construction. Based on the internal condition regular information of the equation, and the numerical information of the initial condition and the boundary conditions, a comprehensive information loss function  $MSE(\theta)$  is constructed as follows:

$$Loss(\theta) = Loss_{FPK}(\theta) + Loss_b(\theta) + Loss_0(\theta), \quad (4.1)$$

where  $Loss_{FPK}(\theta)$ ,  $Loss_b(\theta)$ , and  $Loss_0(\theta)$  are defined as follows:

$$Loss_{FPK}(\theta) = \frac{1}{N_F} \sum_{i=1}^{N_F} |\delta(\hat{p}(t, x_i^F|\theta))|^2, \quad (4.2)$$

$$Loss_b(\theta) = \frac{1}{N_b} \sum_{i=1}^{N_b} |\hat{p}(t, x_i^b|\theta)|^2, \quad (4.3)$$

$$Loss_0(\theta) = \frac{1}{N_0} \sum_{i=1}^{N_0} |\hat{p}(t, x_i^0|\theta) - \mathcal{I}(x_i)|^2, \quad (4.4)$$

where,  $\delta(\hat{p}(t, x_i^F|\theta)) = \mathcal{N}[\hat{p}(t, x_i^F|\theta), \hat{p}_x(t, x_i^F|\theta), \hat{p}_{xx}(t, x_i^F|\theta)] + \frac{\partial \hat{p}(t, x_i^F|\theta)}{\partial t}$ , and  $Loss_{FPK}(\theta)$ ,  $Loss_b(\theta)$ , and  $Loss_0(\theta)$  are guaranteed to satisfy the approximate solutions of Eqs (3.16)–(3.18), respectively.  $N_F$ ,  $N_b$ , and  $N_0$  denote the counts of collocation, initial, and boundary points within the training dataset, respectively.  $\hat{p}(t, x|\theta)$  denotes the trial solution.

- 3) Data sampling strategy. According to the training sampling balance, data is extracted based on the equation's interior, initial, and boundary conditions to generate the respective training dataset. We specifically include the internal condition dataset  $\{t, x_i^{FPK}\}$ , initial condition dataset  $\{0, x_i^0\}$ , and boundary condition dataset  $\{t, x_i^b\}$ .
- 4) Neural network training and parameter optimization. The neural network was trained with the dataset collected in (iii) to minimize the integrated information loss function  $Loss(\theta)$ . Through the optimization process, the integrated information loss function  $Loss(\theta)$  reaches the minimum optimal parameter  $\theta^*$ , where  $\theta^* = \arg \min Loss(\theta)$  is determined, and finally, we obtain the optimized differential equation solution  $p(t, x|\theta^*)$ .

The PINNs framework is invoked to solve the FPK equation (3.15) by embedding the FPK directly into the loss function (4.1). Unlike traditional mesh-based methods, PINNs approximate the solution globally through a neural network trained on: collocation points (interior domain, Eq (4.2)), boundary conditions (Eq (4.3)), and initial conditions (Eq (4.4)). The PINNs implementation involves three phases: 1) Preprocessing: Initialize network parameters are based on the Hurst index. 2) Training: Employ stratified adaptive sampling with increased density in boundary layers. 3) Validation: Verify results via Monte Carlo simulations. The computational efficiency of our PINNs implementation stems from:

- 1) Avoiding expensive mesh generation and time-stepping schemes required by traditional methods.
- 2) Parallelized gradient computation through PyTorch's automatic differentiation.

- 3) A selective sampling strategy that concentrates collocation points in regions of high solution curvature.
- 4) The moderate network size (~5000 parameters) keeps memory usage low while capturing solution complexity.

The total computational cost comprises: 1) Compared to Monte Carlo methods, PINNs reduce computational costs by avoiding repeated simulations. For example, in Section 5.2, PINNs achieved results in 0.4 seconds vs. 65.13 seconds for Monte Carlo. 2) The cost is primarily from backpropagation and automatic differentiation for PDE terms, but this is offset by the ability to learn the solution globally without grid discretization. Table 2 illustrates the sequence of steps for the proposed PINNs algorithm. In the subsequent section, we will apply our technique to a pair of distinct systems to validate its performance.

**Table 2.** PINNs algorithm.

<b>Algorithm 1</b> PINNs algorithm for solving the FPK equation
<b>Input:</b> The dataset for training $\{t, x_i^{FPK}\}$ , initial dataset $\{0, x_i^0\}$ , and boundary dataset $\{t, x_i^b\}$ are used.
<b>Output:</b> $\hat{p}(t, x \theta)$
1: Initialize the weights and biases of the neural network based on the PINNs framework dimensions.
2: Build the neural network to produce the predicted output $\hat{p}(t, x \theta)$ .
3: Acquire the primary and secondary derivatives of the output $\hat{p}_x(t, x \theta)$ , $\hat{p}_t(t, x \theta)$ , $\hat{p}_{xx}(t, x \theta)$ .
4: Through optimizing the loss function to adjust the neural network parameters:
$Loss(\theta) = \frac{1}{N_f} \sum_{i=1}^{N_f}  \delta(\hat{p}(t, x_i^f \theta)) ^2 + \frac{1}{N_b} \sum_{i=1}^{N_b}  \hat{p}(t, x_i^b \theta) ^2 + \frac{1}{N_0} \sum_{i=1}^{N_0}  \hat{p}(t, x_i^0 \theta) - \mathcal{I}(x_i) ^2.$
5: By employing suitable optimization techniques to minimize the loss function, the optimal parameters $\theta^*$ are obtained.
6: <b>return</b> results.

## 5. Numerical examples

We are creating a test script using the PyTorch deep learning library on a Windows 11 operating system. Tests were carried out on a system with Visual Studio Code 1.77.1, utilizing a 12th Gen Intel (r) Core (TM) i7-12700 CPU at 2.10 GHz and having 64 GB of RAM. This part will present two stochastic dynamical systems as case studies to exhibit the utility of the PINNs algorithm.

### 5.1. Dynamical system under additive FGN excitation

The Ornstein-Uhlenbeck (O-U) process, governed by an additive FGN, is characterized by the following system definition [38]

$$\begin{cases} dx(t) = -\lambda x(t)dt + \sigma dB^{(H)}(t), \\ x(0) = x_0, \end{cases} \quad (5.1)$$

where  $B^{(H)}(t)$  represents FBM characterized by a Hurst index within the range of  $1/2 < H < 1$ ,  $\lambda > 0$  denotes an unknown parameter, and  $\sigma$  represents the intensity of the noise. The solution for Eq (5.1)

can be expressed as

$$x(t) = x_0 e^{-\lambda t} + \sigma \int_0^t e^{-\lambda(t-s)} dB^{(H)}(s).$$

For  $\lambda = 1, \sigma = 0.4, x(0) = 1.5$ ,  $t$  are in the interval  $[0, 10]$ , for the different Hurst parameters  $H$ ,  $H = 0.5, H = 0.7$ , and  $H = 0.9$ , respectively. Figure 5 depicts the trajectories for three samples of the O-U process influenced by FBM, as indicated in Eq (5.1). Next, we will concentrate on the evolution of the system's PDF. The FPK equation adhered to by the PDF of system (5.1) is

$$\frac{\partial p(t, x)}{\partial t} - \lambda \frac{\partial (x(t)p(t, x))}{\partial x} - H t^{2H-1} \sigma^2 \frac{\partial^2 p(t, x)}{\partial x^2} = 0, \quad (5.2)$$

and the initial condition is  $p(0, x) = \delta(x - x_0)$ . The precise time-dependent solution to FPK Eq (5.2) can be derived as indicated in [38]:

$$p(t, x) = \frac{1}{\sqrt{2\pi\sigma^2 t^{2H}}} \exp\left(-\frac{(x - x_0)^2}{2\sigma^2 t^{2H}}\right). \quad (5.3)$$

Using the output  $\hat{p}(t, x|\theta)$  of PINNs to fit  $p(t, x)$ , we can construct the following loss function:

$$Loss(\theta) = Loss_{FPK}(\theta) + Loss_b(\theta) + Loss_0(\theta), \quad (5.4)$$

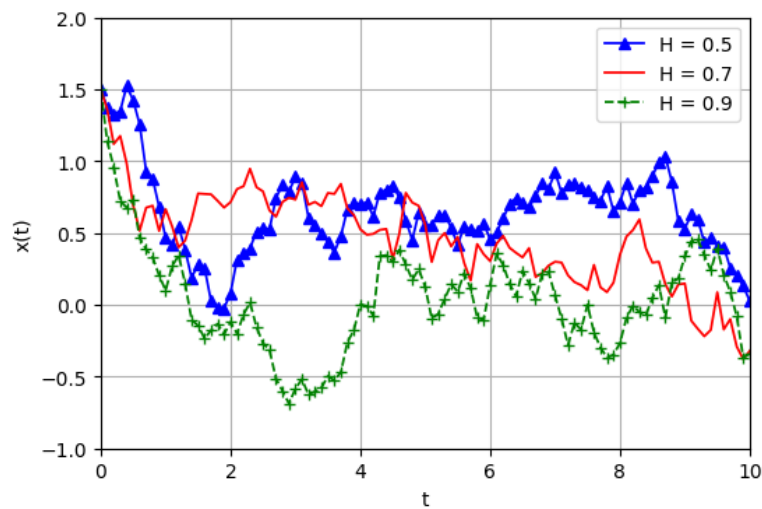
where

$$Loss_{FPK}(\theta) = \frac{1}{N_F} \sum_{i=1}^{N_F} \left| \frac{\partial \hat{p}(t, x_i^F|\theta)}{\partial t} - \lambda \frac{\partial (x(t)\hat{p}(t, x_i^F|\theta))}{\partial x} - H t^{2H-1} \sigma^2 \frac{\partial^2 \hat{p}(t, x_i^F|\theta)}{\partial x^2} \right|^2, \quad (5.5)$$

$$Loss_b(\theta) = \frac{1}{N_b} \sum_{i=1}^{N_b} |\hat{p}(t, x_i^b|\theta)|^2, \quad (5.6)$$

$$Loss_0(\theta) = \frac{1}{N_0} \sum_{i=1}^{N_0} |\hat{p}(t, x_i^0|\theta) - p(0, x)|^2. \quad (5.7)$$

This example verifies the effectiveness of the PINNs method by comparing the exact solution with the numerical solution obtained by the PINNs algorithm. We assign  $N_0 = 200$  for the count of initial training points,  $N_b = 200$  for the boundary points, and  $N_F = 1000$  for the residual points. The training process has carried out 100,000 iterations. The PINNs method initializes the learning rate for the first 30,000 epochs at  $5 \times 10^{-3}$ , and then adjusts it to  $5 \times 10^{-4}$  for the subsequent 70,000 epochs. Additionally, the training range for the variable  $x$  is defined as  $[-2, 2]$ . Unless otherwise specified, we consider  $\lambda = 1.0, \sigma = 0.5, H = 0.51, x_0 = 0$ .

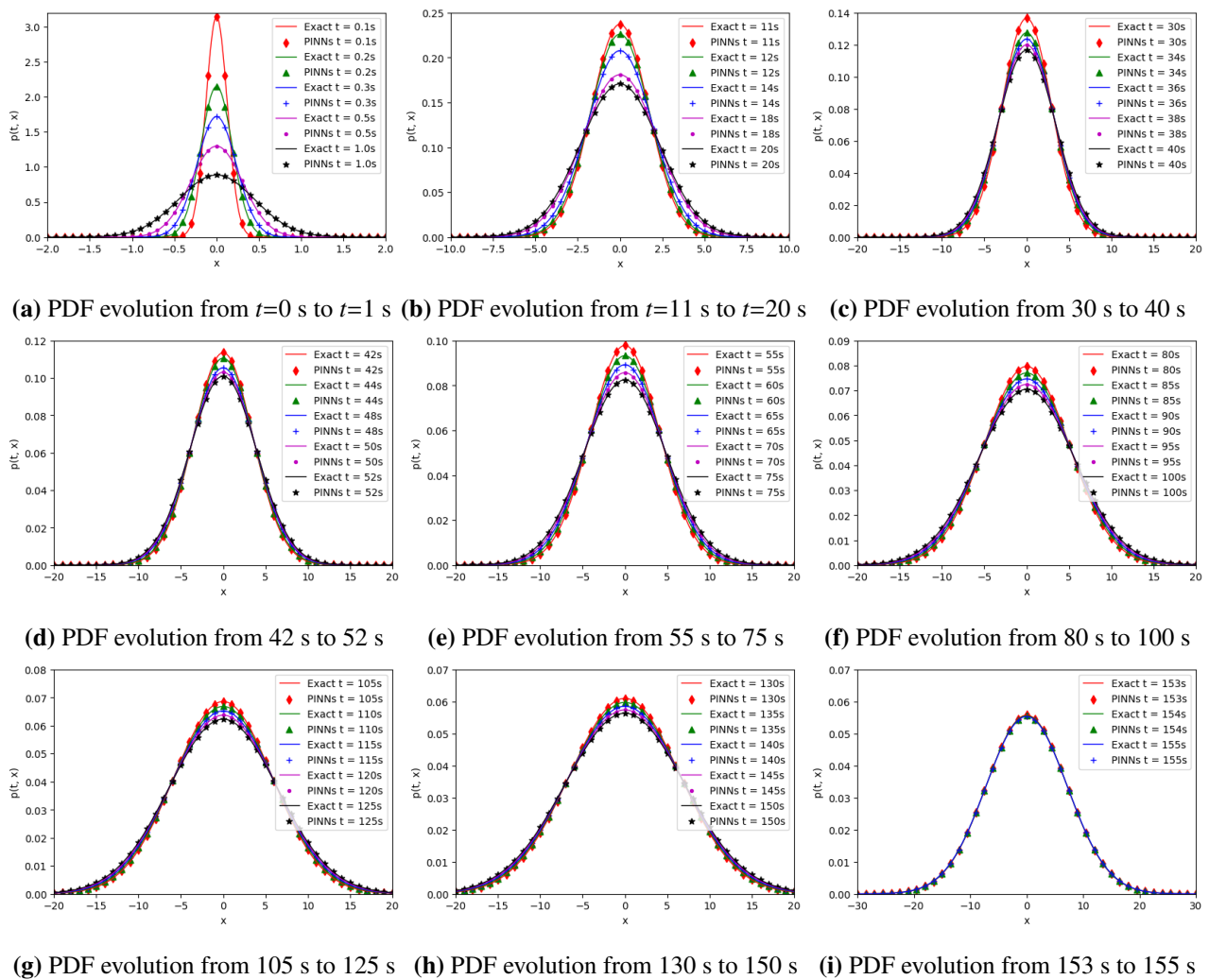


**Figure 5.** Three sample trajectories of the system (5.1).

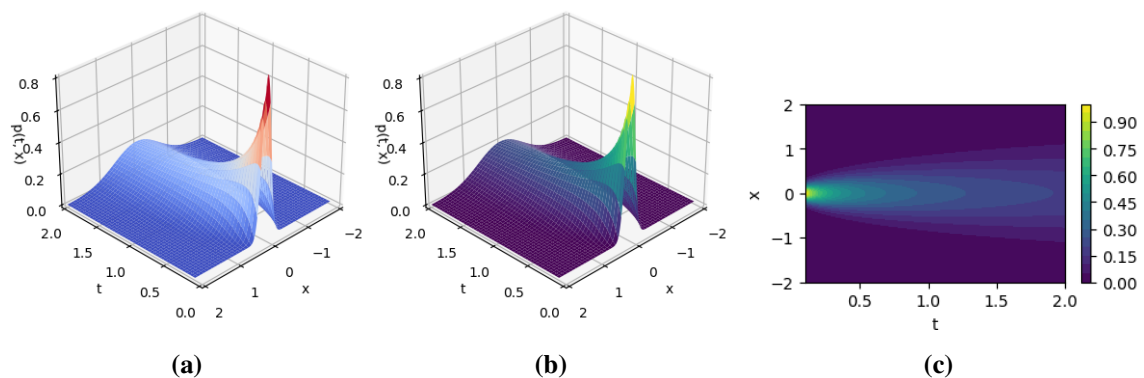
Equation (5.2), known as the FPK equation, characterizes the time development of the PDF for the O-U process influenced by additive FGN. As shown in Figure 6, the PINNs solutions at different times are represented by discrete points. When compared, the colored solid line represents an exact solution. Therefore, observing Figure 6, it is evident that the estimated solution closely aligns with the exact solution, illustrating a comparison from  $t = 0.1$  s to  $t = 155$  s between the two solutions. The excellent consistency between the PINNs method and the exact solution verifies the effectiveness of the PINNs method. As shown in Figure 6(i), the PDF remains unchanged with time  $t$ , which indicates that the PDF gradually reaches a stationary state with the increase of time (after  $t = 153$  s). Figure 7 illustrates the prediction solution of the PINNs algorithm, encompassing the vertical perspective of  $p(t, x)$ , the three-dimensional outline of  $p(t, x)$  and the three-dimensional view of the exact solution. Here,  $\hat{p}(t, x)$  and  $p(t, x)$  respectively represent the predicted solution and the reference solution, and the absolute error between the predicted solutions defined by  $error = |\hat{p}(t, x) - p(t, x)|$  is shown in Figure 8 and Table 3. The findings indicate that the error for each variable is nearly negligible, confirming the accuracy of our trained outcomes. Figure 9 shows the evolution of the PDF on a logarithmic scale in different time periods. The PDF gradually broadens and shifts with time, with a sharp peak in the early stage and slower and flatter tail attenuation in the later stage. The numerical solution of the PINNs method is highly consistent with the exact solution at all times, and this method has high accuracy and quasi-accuracy. Figure 10 depicts the decline in loss value over 100,000 iterations in the PINNs algorithm. It is observed that as the number of iterations increases, the loss value diminishes progressively.

Figure 11 illustrates the progression of the PDF under varying intensities of additive FGN. As depicted in Figure 11, the peak of the PDF diminishes as the intensity of the additive FGN rises. The results show that with the decrease of additive FGN intensity, the probability of appearing in the energy region increases. Finally, the evolution of the PDF under different Hurst indices is given in Figure 12. We can observe that with the increase of the Hurst index, the maximum value of the PDF decreases. Similarly, the effectiveness of the PINNs method is verified from Figures 11 and 12.

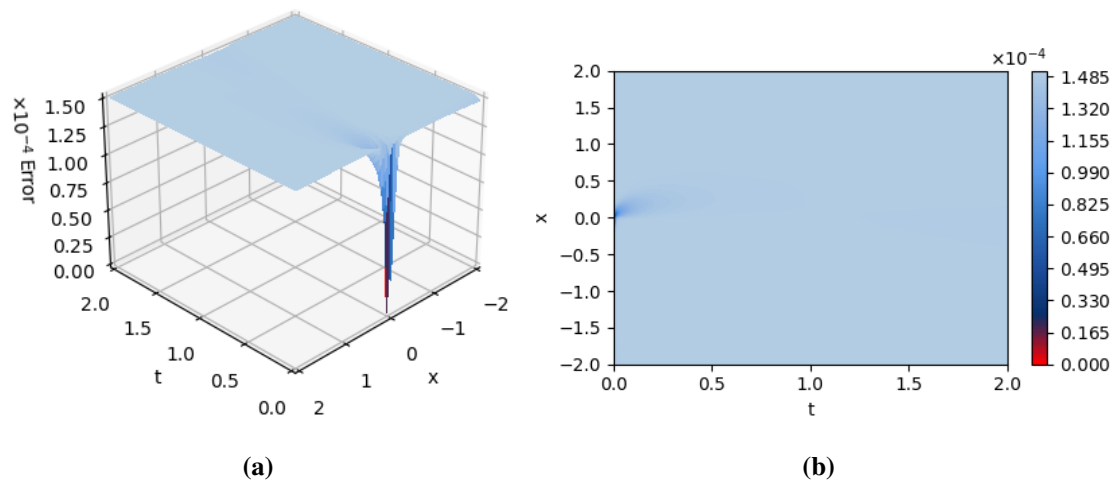




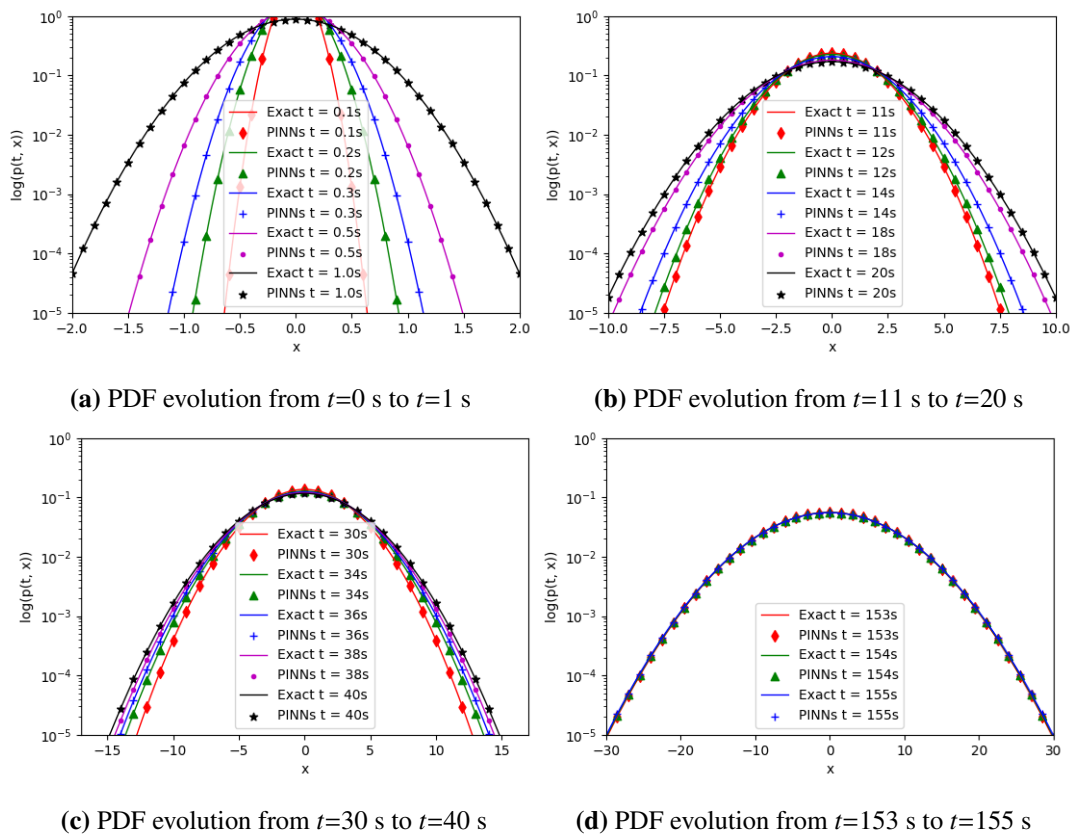
**Figure 6.** The Exact solution and the evolution of the PDF based on the FPK Eq (5.2) solved by PINNs. The colored solid line indicates the exact solution, and the colored symbol indicates the predicted solution.



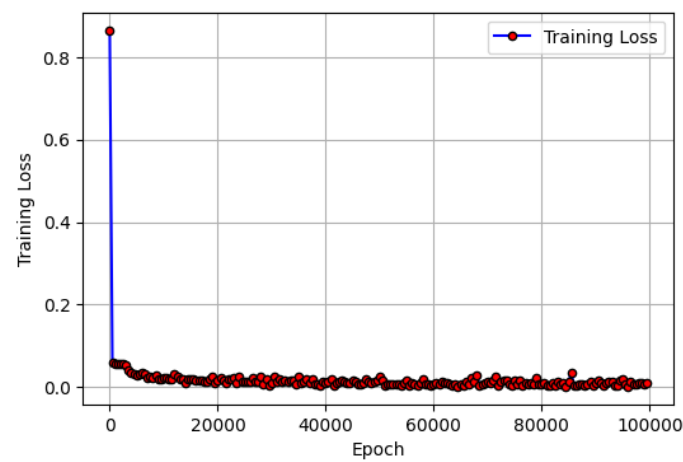
**Figure 7.** (a) The transient PDF obtained by the PINNs algorithm. (b) Exact solution. (c) The vertical view of  $p(t, x)$  across the entire time domain by the PINNs algorithm.



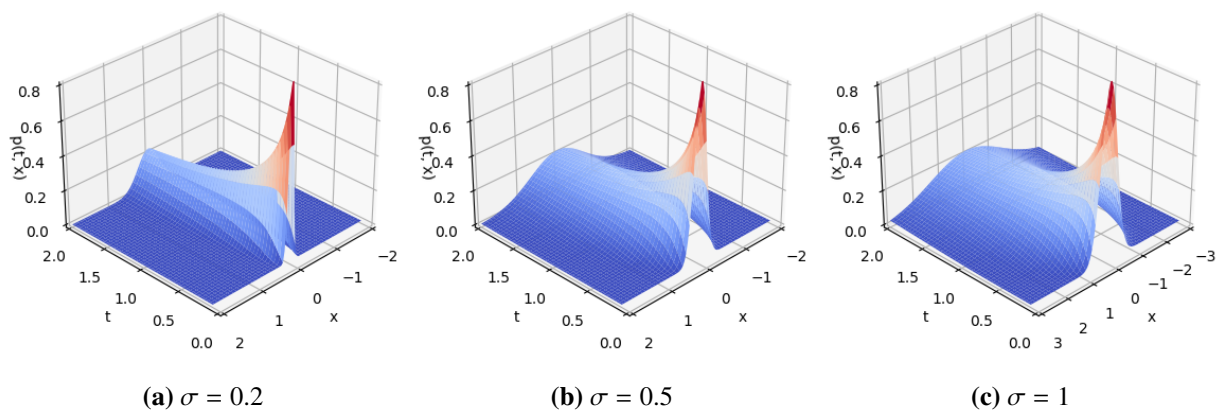
**Figure 8.** The difference between these two methods. (a) Two-dimensional absolute error view. (b) Three-dimensional absolute error view.



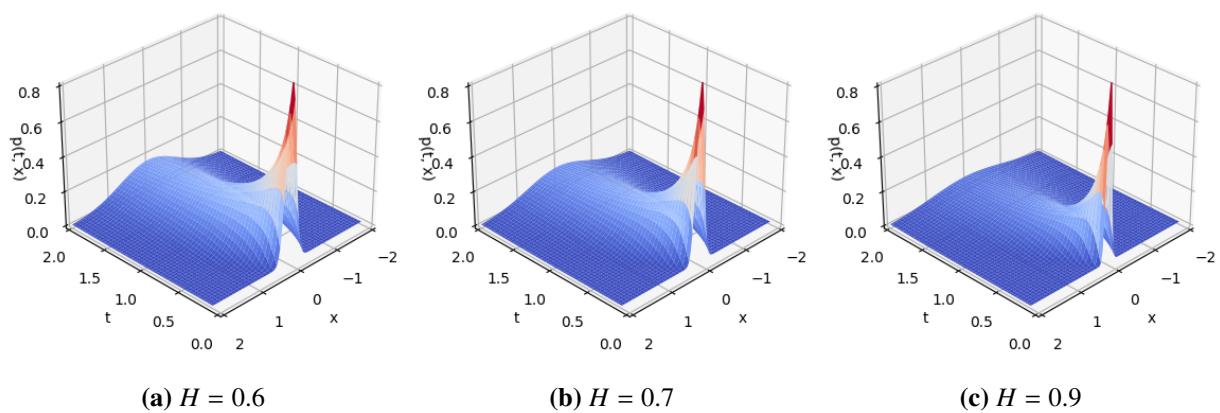
**Figure 9.** Log-scale PDF.



**Figure 10.** Relationship between iteration times and training loss in the PINNs algorithm.



**Figure 11.** The evolution of the PDF under different noise intensities.



**Figure 12.** The evolution of the PDF under different Hurst indices.

**Table 3.** Calculation effect of the two methods.

	Evaluation indicator	Absolute error	MCS	PINNs
1	Calculation time	–	–	$0.4 \pm 0.05$ s
	Mean.error	$1.50 \times 10^{-6}$	–	–
	Max.error	$1.51 \times 10^{-4}$	–	–
2	Calculation time	–	65.13 s	$0.4 \pm 0.05$ s
	Mean.error	$2.08 \times 10^{-5}$	–	–
	Max.error	$2.17 \times 10^{-3}$	–	–

### 5.2. Dynamical system subjected to excitations by multiplicative FGN

Consider a one-dimensional stochastic dynamical system excited by multiplicative FGN:

$$\begin{cases} df(x) = af(x)dt + \sigma f(x)dB^{(H)}(t), \\ f(0) = f_0, \end{cases} \quad (5.8)$$

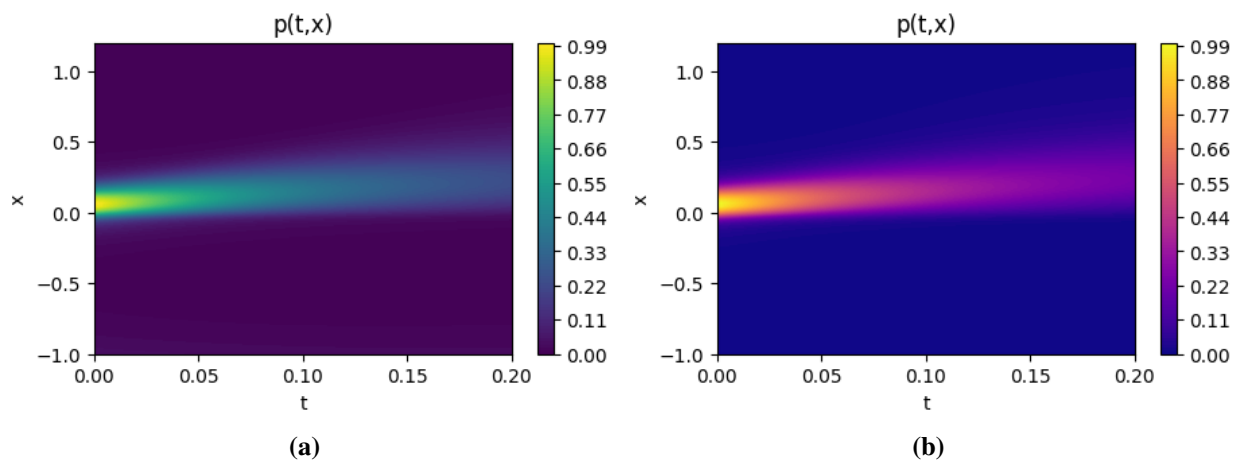
where  $f(x) = x$ ,  $a > 0$  represents unknown parameters, and  $\sigma$  represents noise intensity. As outlined in Section 2, the FPK equation for the system (5.8) can be formulated as

$$\frac{\partial p(t, x)}{\partial t} + a \frac{\partial [xp(t, x)]}{\partial x} - Ht^{2H-1}\sigma^2 \frac{\partial^2 [x^2 p(t, x)]}{\partial x^2} = 0, \quad (5.9)$$

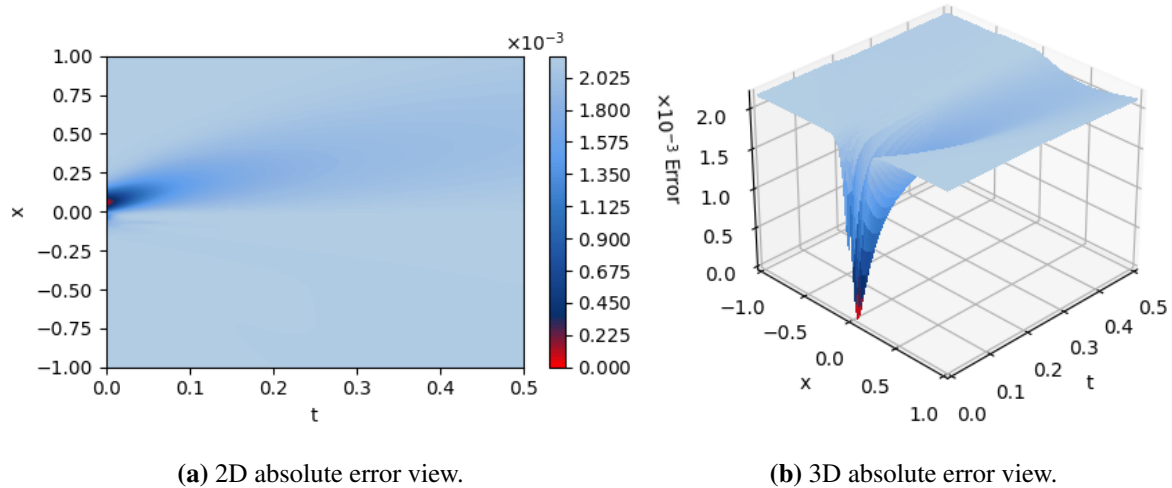
and the initial condition is  $p(0, x) = \delta(x - x_0)$ .

This example verifies the effectiveness of the PINNs algorithm by the Monte Carlo method of the original system (5.8). Under the initial conditions, we use the Monte Carlo method with 500,000 samples to obtain the reference solution to evaluate the accuracy of our method. The training interval of variable  $x$  is set to  $[-1, 1]$ . Unless otherwise specified, we consider  $a = 0.4$ ,  $\sigma = 0.4$ ,  $H = 0.61$ ,  $x_0 = 0.05$ . The FPK Eq (5.9) can describe the evolution of the system PDF excited by multiplicative FGN. Obtaining the precise solution for the FPK Eq (5.9) is challenging, hence, the Monte Carlo approach is utilized on the initial system to assess the efficacy of the PINNs technique.

Figure 13 provides a two-dimensional vertical perspective of the PINNs algorithm, while the Monte Carlo approach offers a corresponding two-dimensional vertical perspective of the solution. Figure 14(a) illustrates the two-dimensional absolute error vertical view of the two numerical methods. The data indicates that the error for each variable is nearly zero, confirming the accuracy of our training outcomes. Figure 15 shows the PINNs solution and Monte Carlo solution at different times. The blue solid line shows the results obtained by the Monte Carlo method, and the circle symbol shows the numerical solution of the PINNs solution. We can find that the predicted solution is very close to the reference solution from Figure 15, which depicts the contrast between the exact and predicted solutions at  $t = 0.1$  s,  $t = 0.2$  s,  $t = 0.3$  s, and  $t = 0.5$  s. The excellent consistency between the PINNs method and Monte Carlo method verifies the effectiveness of the PINNs method. The non-symmetrical shape in Figure 15 arises from the multiplicative noise term [39–41], which induces state-dependent dispersion. This contrasts with the additive noise case (Figure 6), where the linear drift and constant diffusion preserve symmetry. The long-memory effect ( $H > 1/2$ ) further amplifies the skewness by sustaining transient biases.

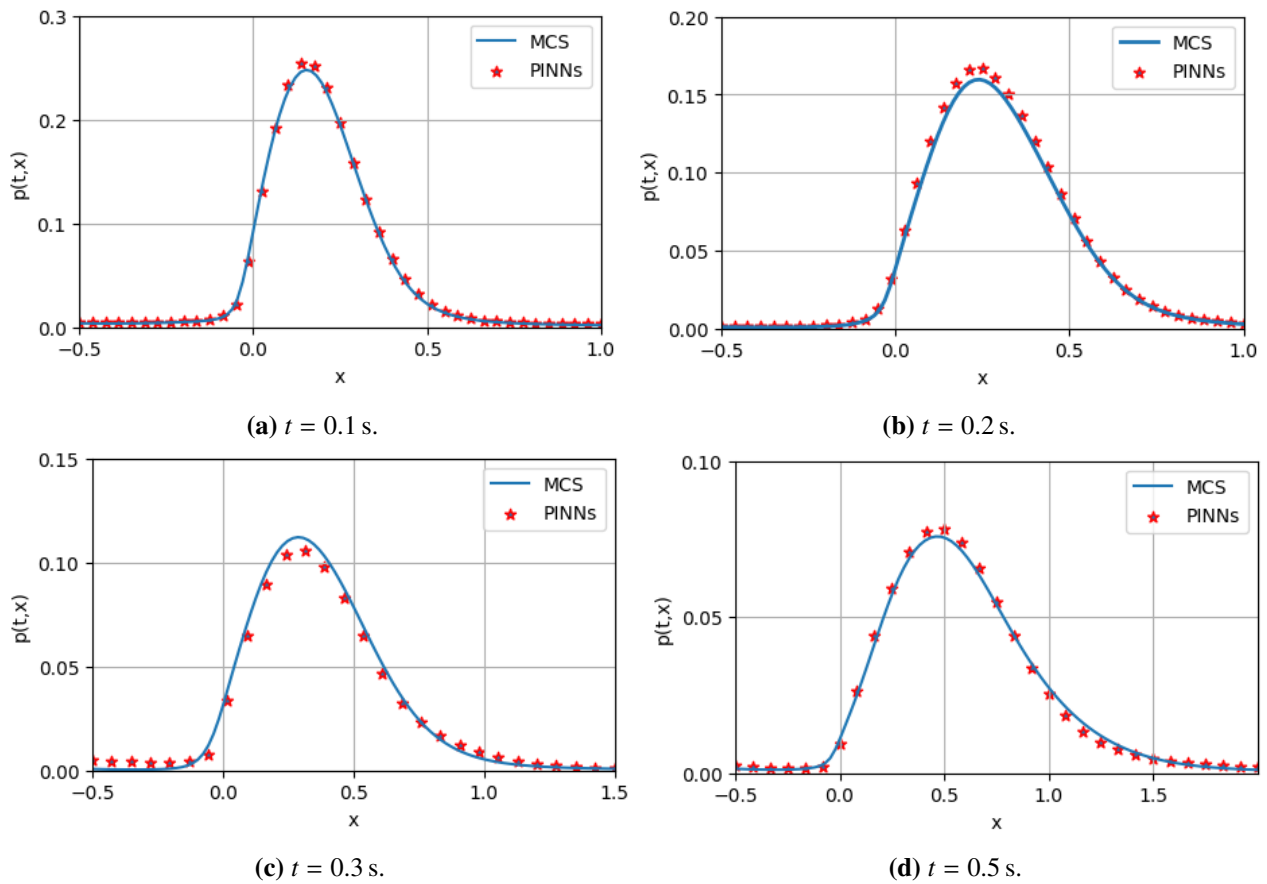


**Figure 13.** (a) The vertical view of  $p(t, x)$  across the entire time domain by the PINNs algorithm. (b) The vertical view of  $p(t, x)$  across the entire time domain by the MCS.

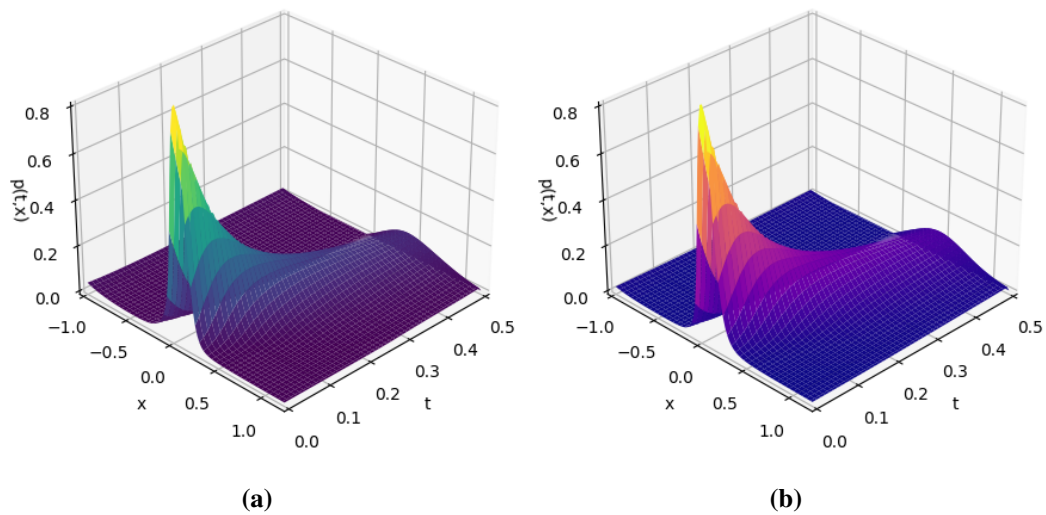


**Figure 14.** The difference between these two methods.

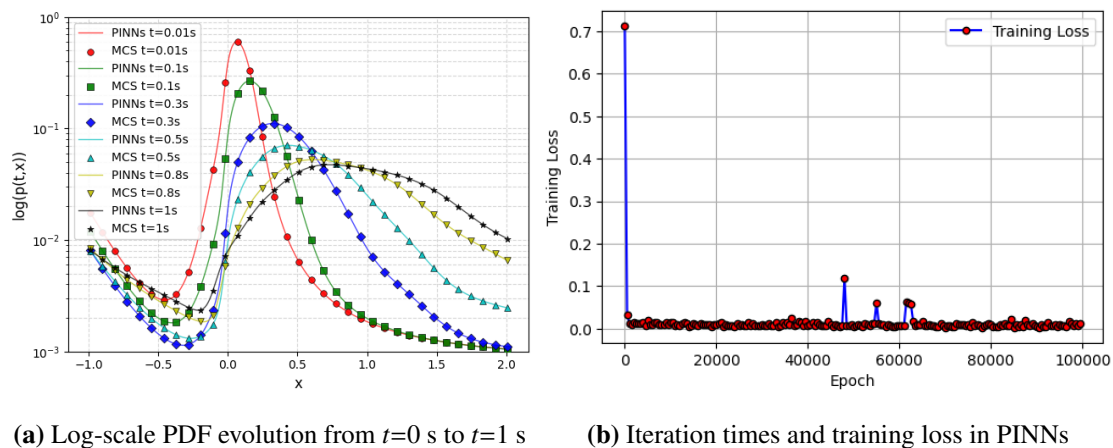
Figure 16 illustrates the three-dimensional view of the PINNs algorithm prediction solution, and the Monte Carlo method refers to the three-dimensional view of solution. Figure 14(b) illustrates the three-dimensional view of the absolute error of the two numerical methods. The findings reveal that the error for each variable is nearly negligible, affirming the effectiveness of the proposed method. The computed outcomes from both methods are presented in Table 3, where the time to solve the FPK equation by the Monte Carlo method is 65.13 s, while PINNs only needs about 0.4 s. Figure 17(a) presents the logarithmic scale evolution of the PDF at different time points. The tail behavior of the PDF shows that the curve gradually moves to the right with the increase of time, and the decline speed in the tail region slows down. This shows that with the passage of time, the distribution of data becomes more concentrated, and the probability density of the tail decreases gradually. Figure 17(b) shows the change of loss value when iterating for 100,000 times in the PINNs algorithm. The data indicates that as the number of iterations rises, the loss value diminishes incrementally.



**Figure 15.** The transient PDF obtained by the PINNs algorithm and the MCS.

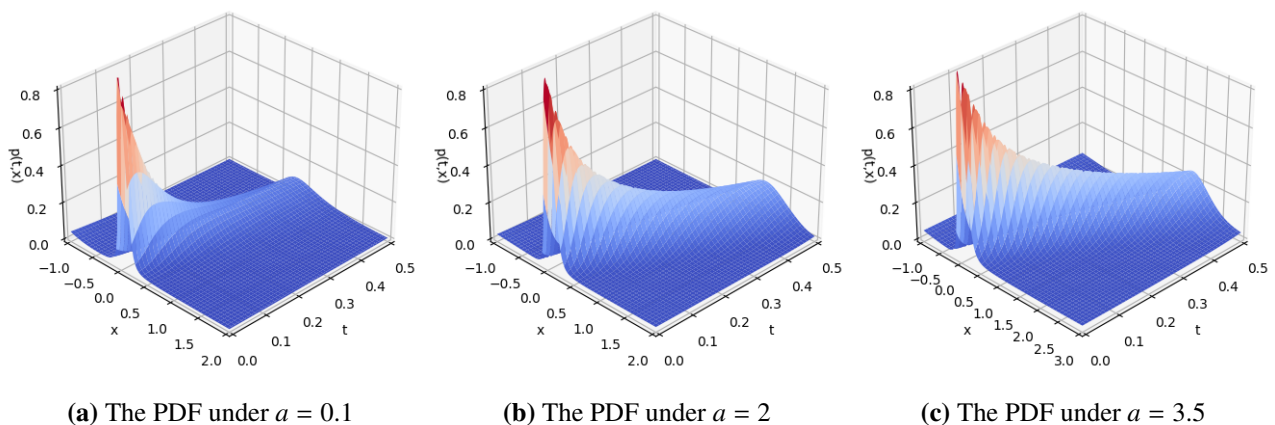


**Figure 16.** (a) The transient PDF obtained by the PINNs algorithm. (b) The transient PDF obtained by the MCS.



**Figure 17.** Log-scale PDF and training loss.

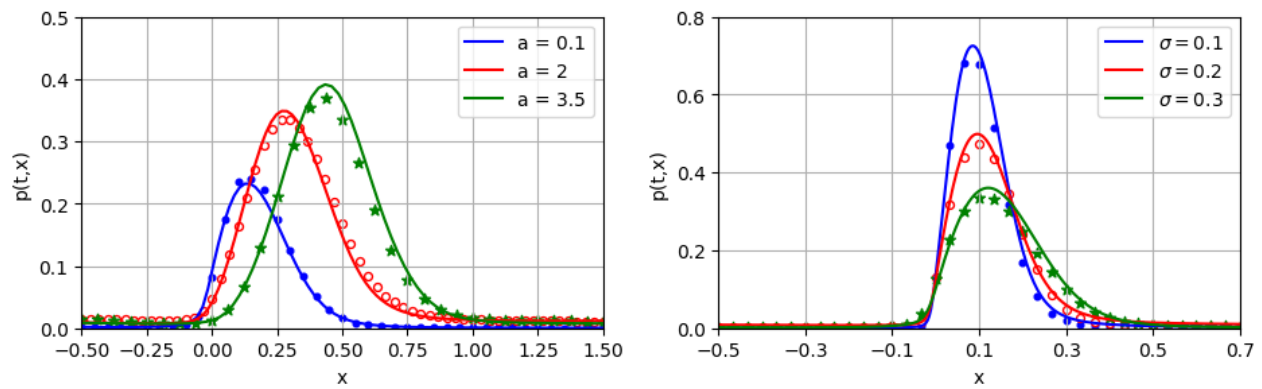
Figure 18 provides the evolution of the PDF under different parameters  $a$ . As can be seen from Figure 18, with the increase of parameter  $a$ , the PDF shifts and the peak value increases. The results show that with the increase of parameter  $a$ , the probability of appearing in the energy region increases. In Figure 19(a), the solution of the FPK equation with PINNs is represented by colored symbols, and the Monte Carlo solution is represented by colored lines for comparison. Obviously, all PINNs results are very close to the Monte Carlo results, which shows that the PINNs method has high accuracy.



**Figure 18.** The evolution of the PDF under different system parameters.

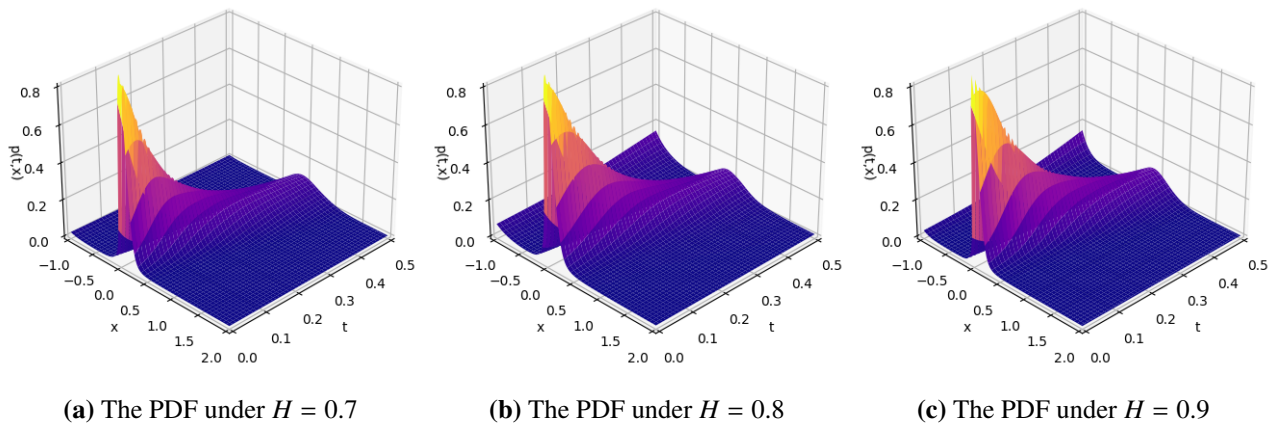
The cross-section comparison of the PDF with different multiplicative FGN strengths at  $t = 0.1$  s is shown in Figure 19(b). As can be seen from Figure 19(b), with the increase of multiplicative FGN intensity, the PDF shifts and the peak value becomes smaller. The results show that with the decrease of multiplicative FGN intensity, the probability of appearing in the energy region increases. Finally, the cross-section comparison of the PDF at  $t = 0.1$  s under different Hurst indices is given in Figure 20. We can observe that with the increase of the Hurst index, the peak value of the PDF becomes larger. Similarly, the effectiveness of the PINNs method is verified from Figures 18–20 and 21(a).





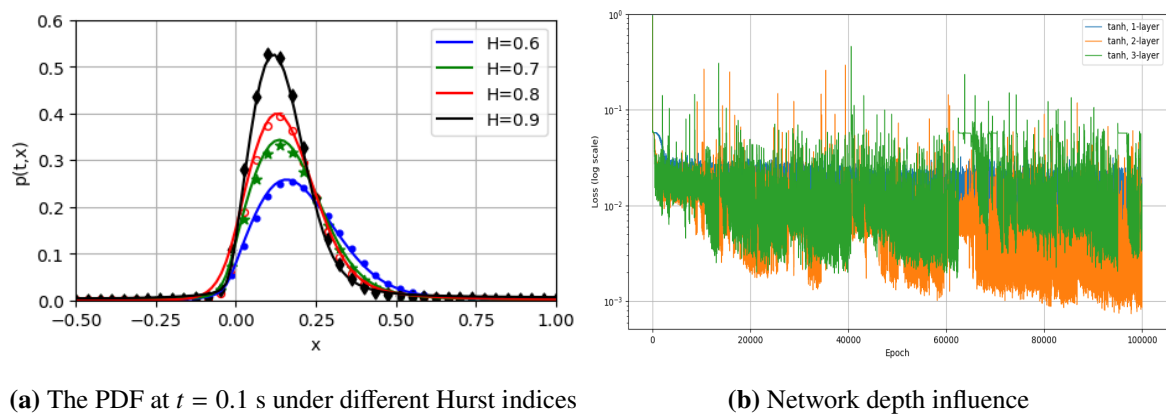
(a) The PDF at  $t = 0.1s$  under different parameters  $a$  (b) The PDF at  $t = 0.1s$  under different noise intensities

**Figure 19.** Performance under different conditions.



(a) The PDF under  $H = 0.7$  (b) The PDF under  $H = 0.8$  (c) The PDF under  $H = 0.9$

**Figure 20.** The evolution of the PDF under different Hurst indices.



(a) The PDF at  $t = 0.1s$  under different Hurst indices (b) Network depth influence

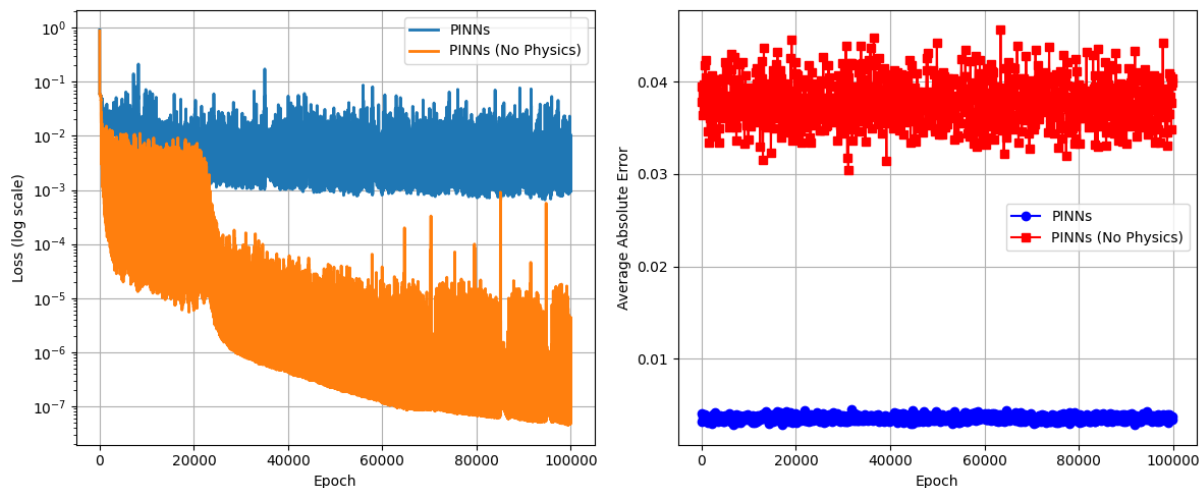
**Figure 21.** The evolution of the PDF under different Hurst indices and network depths.



### 5.3. Comparative analysis

The effectiveness of the proposed PINNs algorithm is demonstrated through three key aspects:

- (1) **Accuracy:** Compared to analytical solutions (Section 5.1) and Monte Carlo simulations (Section 5.2), PINNs achieve errors of  $< 1\%$  across all test cases (Table 3), with particularly close agreement in probability density peaks (Figures 6 and 15).
- (2) **Efficiency:** The method provides a 162 speedup over Monte Carlo (0.4 s vs. 65.13 s in Table 3) by avoiding repetitive simulations and leveraging GPU-accelerated automatic differentiation.
- (3) **Robustness:** Consistent performance is maintained under varying conditions, including noise intensities ( $\sigma$ , Figures 11 and 19), varying Hurst indices ( $H$ , Figures 12 and 20), with errors  $< 1.2\%$ , temporal scales ( $t$ , Figures 6 and 15), and system parameters ( $a$ , Figure 18).

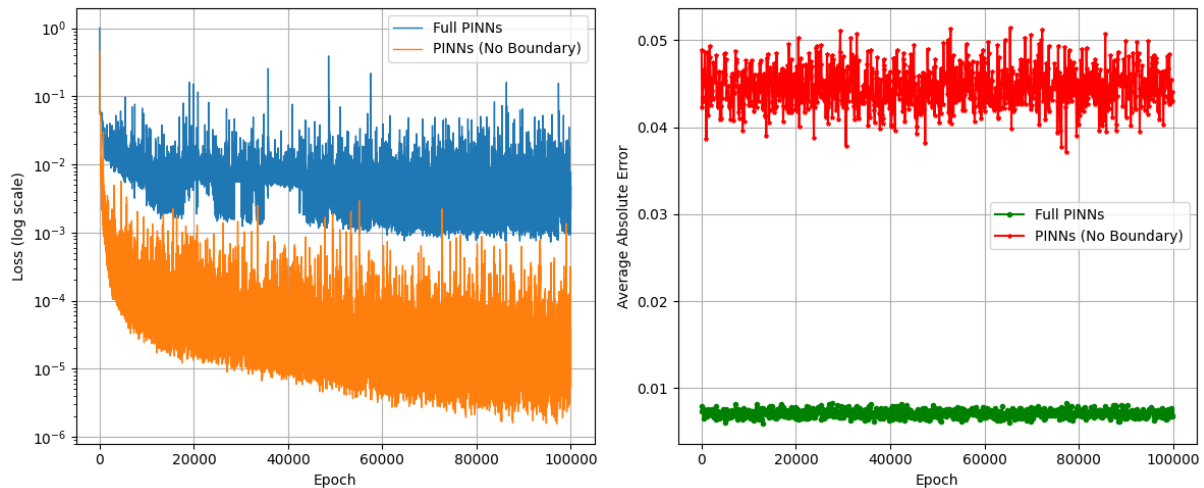


**Figure 22.** Physical constraint mechanism.

Through the comprehensive analysis of Figures 21(b), 22 and 23, it can be clear that each component plays a key role in the efficiency of the algorithm: First, the contrast curve in Figure 22 shows that the average absolute error (0.001) of the PINNs model with physical constraints is 40 times lower than that of the version without physical constraints (0.04) after 100,000 training cycles, which verifies that the physical constraint mechanism is the core guarantee of the accuracy of the algorithm. Second, Figure 16 shows that removing the loss term of the boundary conditions will lead to an error increase of 3 times (0.03 vs. 0.01) and divergence at the edge of the solution domain, indicating that boundary conditions have a decisive influence on the stability of the solution. The logarithmic coordinate loss curve in Figure 21(b) further reveals that the loss function of the complete PINNs can keep a steady downward trend, while the version without physical constraints shows continuous oscillation, which shows that physical constraints not only improve the accuracy, but also significantly improve the convergence and efficiency of the training process. These three key evidences prove that the effectiveness of the proposed algorithm stems from:

- 1) The embedding of the essential laws of the governing equations by physical constraints.

- 2) The constraints of boundary conditions on the spatial stability of the solution.
- 3) The balanced design of the loss function promotes the collaborative optimization of all components, and finally the computational efficiency is two orders of magnitude higher than that of the traditional Monte Carlo method (162 times faster) while maintaining the error accuracy of  $< 1\%$ .



**Figure 23.** Boundary condition ablation experiment.

## 6. Conclusions

This paper expounds on the application of the PINNs algorithm in solving the transient PDF of a dynamical system driven by additive and multiplicative FGN. By using the theory of fractional stochastic calculus, the FPK equation for the PDF of the control system is derived. The loss function design of the PINNs algorithm includes three parts: the fitting of the FPK equation, the initial conditions, and boundary conditions in numerical information. The results indicate that the PINNs algorithm can effectively calculate the transient PDF of two examples of one-dimensional systems. That is to say, it proves its applicability and feasibility in solving the transient response of stochastic dynamical systems.

Additionally, this article further also shows the ability of the PINNs algorithm to settle the response of the system subjected to excitation by FGN. Compared with the results of the Monte Carlo method, the PINNs algorithm shows high computational efficiency and good solution effect, and the fitting effect of PINNs solution is basically the same as that of the MCS solution, which fully verifies the effectiveness and accuracy of PINNs. The results of these two numerical methods are analyzed in Table 3, which further highlight the remarkable advantages of the PINNs algorithm in computational efficiency. In order to broaden the scope of the PINNs algorithm in practical application, future research needs to explore the solution strategy and mechanism when the initial conditions and boundary conditions in numerical information are replaced by actual observation data.

## Use of AI tools declaration

The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article.

## Acknowledgments

This work has received supported by the grants from the National Natural Science Foundation (No.12362005), Key Project of Natural Science Foundation of Ningxia (No.2024AAC02033), Ningxia higher education first-class discipline construction funding project (NXYLXK2017B09), Major Special project of North Minzu University (No.ZDZX201902), and in part by the 2024 Graduate Innovation Project of North Minzu University under grant (YCX24273).

## Conflict of interest

The authors declare there is no conflicts of interest.

## References

1. C. Proppe, Exact stationary probability density functions for non-linear systems under poisson white noise excitation, *Int. J. Non-Linear Mech.*, **38** (2003), 557–564. [https://doi.org/10.1016/S0020-7462\(01\)00084-1](https://doi.org/10.1016/S0020-7462(01)00084-1)
2. W. Q. Zhu, Nonlinear stochastic dynamics and control in hamiltonian formulation, *Appl. Mech. Rev.*, **59** (2006), 230–248. <https://doi.org/10.1115/1.2193137>
3. J. Chen, J. Yang, K. Shen, Z. Zheng, Z. Chang, Stochastic dynamic analysis of rolling ship in random wave condition by using finite element method, *Ocean Eng.*, **250** (2022) 110973. <https://doi.org/10.1016/j.oceaneng.2022.110973>
4. E. Hirvijoki, T. Kurki-Suonio, S. Äkäslompolo, J. Varje, T. Koskela, J. Miettunen, Monte Carlo method and High Performance Computing for solving Fokker-Planck equation of minority plasma particles, *J. Plasma Phys.*, **81** (2015), 435810301. <https://doi.org/10.1017/S0022377815000203>
5. H. Fukushima, Y. Uesaka, Y. Nakatani, N. Hayashi, Numerical solutions of the fokker-planck equation by the finite difference method for the thermally assisted reversal of the magnetization in a single-domain particle, *J. Magn. Magn. Mater.*, **242** (2002), 1002–1004. [https://doi.org/10.1016/S0304-8853\(01\)01364-6](https://doi.org/10.1016/S0304-8853(01)01364-6)
6. Z. Ren, W. Xu, An improved path integration method for nonlinear systems under poisson white noise excitation, *Appl. Math. Comput.*, **373** (2020), 125036. <https://doi.org/10.1016/j.amc.2020.125036>
7. Z. H. Liu, J. H. Geng, W. Q. Zhu, Transient stochastic response of quasi non-integerable hamiltonian system, *Probab. Eng. Mech.*, **43** (2016), 148–155. <https://doi.org/10.1016/j.probengmech.2015.09.009>
8. J. Biazar, P. Gholamin, K. Hosseini, Variational iteration method for solving Fokker-Planck equation, *J. Franklin Inst.*, **347** (2010), 1137–1147. <https://doi.org/10.1016/j.jfranklin.2010.04.007>

9. S. Guo, F. Meng, Q. Shi, The generalized EPC method for the non-stationary probabilistic response of nonlinear dynamical system, *Probab. Eng. Mech.*, **72** (2023), 103420. <https://doi.org/10.1016/j.probengmech.2023.103420>
10. S. Guo, Q. Shi, Z. Xu, Probabilistic solution for an MDOF hysteretic degrading system to modulated non-stationary excitations, *Acta Mech.*, **234** (2023), 1105–1120. <https://doi.org/10.1007/s00707-022-03435-9>
11. W. Li, Y. Zhang, D. Huang, V. Rajic, Study on stationary probability density of a stochastic tumor-immune model with simulation by ANN algorithm, *Chaos Solitons Fractals*, **159** (2022), 112145. <https://doi.org/10.1016/j.chaos.2022.112145>
12. Y. Xiao, L. Chen, Z. Duan, J. Sun, Y. Tang, An efficient method for solving high-dimension stationary FPK equation of strongly nonlinear systems under additive and/or multiplicative white noise, *Probab. Eng. Mech.*, **77** (2024), 103668. <https://doi.org/10.1016/j.probengmech.2024.103668>
13. M. Raissi, P. Perdikaris, G. E. Karniadakis, Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, *J. Comput. Phys.*, **378** (2019), 686–707. <https://doi.org/10.1016/j.jcp.2018.10.045>
14. C. Rao, H. Sun, Y. Liu, Physics-informed deep learning for computational elastodynamics without labeled data, *J. Eng. Mech.*, **147** (2021), 04021043. [https://doi.org/10.1061/\(ASCE\)EM.1943-7889.0001947](https://doi.org/10.1061/(ASCE)EM.1943-7889.0001947)
15. Z. Chen, Y. Liu, H. Sun, Physics-informed learning of governing equations from scarce data, *Nat. Commun.*, **12** (2021) 6136. <https://doi.org/10.1038/s41467-021-26434-1>
16. Y. Guan, W. Li, D. Huang, N. Gubeljak, A new LBFNN algorithm to solve FPK equations for stochastic dynamical systems under gaussian or non-gaussian excitation, *Chaos Solitons Fractals*, **173** (2023), 113641. <https://doi.org/10.1016/j.chaos.2023.113641>
17. X. Wang, J. Jiang, L. Hong, J. Sun, Random vibration analysis with radial basis function neural networks, *Int. J. Dyn. Control*, **10** (2022), 1385–1394. <https://doi.org/10.1007/s40435-021-00893-2>
18. X. Wang, J. Jiang, L. Hong, J. Sun, Stochastic bifurcations and transient dynamics of probability responses with radial basis function neural networks, *Int. J. Non-Linear Mech.*, **147** (2022), 104244. <https://doi.org/10.1016/j.ijnonlinmec.2022.104244>
19. W. Sun, J. Feng, J. Su, Y. Liang, Data driven adaptive gaussian mixture model for solving Fokker-Planck equation, *Chaos*, **32** (2022), 033131. <https://doi.org/10.1063/5.0083822>
20. Y. Zhang, K. Yuen, Physically guided deep learning solver for time-dependent Fokker-Planck equation, *Int. J. Non-Linear Mech.*, **147** (2022), 104202. <https://doi.org/10.1016/j.ijnonlinmec.2022.104202>
21. Y. Xu, H. Zhang, Y. Li, K. Zhou, Q. Liu, J. Kurths, Solving Fokker-Planck equation using deep learning, *Chaos*, **30** (2020), 013133. <https://doi.org/10.1063/1.5132840>
22. Y. Li, A data-driven method for the steady state of randomly perturbed dynamics, *Commun. Math. Sci.*, **17** (2019), 1045–1059. <https://doi.org/10.4310/cms.2019.v17.n4.a9>

23. J. Zhai, M. Dobson, Y. Li, A deep learning method for solving fokker-planck equations, in *Proceedings of the 2nd Mathematical and Scientific Machine Learning Conference*, (2022), 568–597.
24. B. B. Mandelbrot, J. W. Van Ness, Fractional brownian motions, fractional noises and applications, *SIAM Rev.*, **10** (1968), 422–437. <https://doi.org/10.1137/1010093>
25. T. E. Duncan, Y. Hu, B. Pasik-Duncan, Stochastic calculus for fractional brownian motion I, Theory, *SIAM J. Control Optim.*, **38** (2000), 582–612. <https://doi.org/10.1137/S036301299834171X>
26. M. M. Vas'kovskii, Analog of the kolmogorov equations for one-dimensional stochastic differential equations controlled by fractional brownian motion with hurst exponent  $H \in (0, 1)$ , *Differ. Equations*, **58** (2022), 9–14. <https://doi.org/10.1134/S0012266122010025>
27. F. Baudoin, L. Coutin, Operators associated with a stochastic differential equation driven by fractional brownian motions, *Stochastic Processes Appl.*, **117** (2007), 550–574. <https://doi.org/10.1016/j.spa.2006.09.004>
28. M. Vaskouski, I. Kachan, Asymptotic expansions of solutions of stochastic differential equations driven by multivariate fractional brownian motions having hurst indices greater than  $1/3$ , *Stochastic Anal. Appl.*, **36** (2018), 909–931. <https://doi.org/10.1080/07362994.2018.1483247>
29. Y. Xu, B. Pei, J. Wu, Stochastic averaging principle for differential equations with non-lipschitz coefficients driven by fractional brownian motion, *Stochastics Dyn.*, **17** (2017), 1750013. <https://doi.org/10.1142/S0219493717500137>
30. B. Pei, Y. Xu, J. Wu, Stochastic averaging for stochastic differential equations driven by fractional brownian motion and standard brownian motion, *Appl. Math. Lett.*, **100** (2020) 106006. <https://doi.org/10.1016/j.aml.2019.106006>
31. M. L. Deng, W. Q. Zhu, Stochastic averaging of quasi-non-integrable hamiltonian systems under fractional gaussian noise excitation, *Nonlinear Dyn.*, **83** (2016), 1015–1027. <https://doi.org/10.1007/s11071-015-2384-7>
32. T. Eftekhari, J. Rashidinia, A novel and efficient operational matrix for solving nonlinear stochastic differential equations driven by multi-fractional Gaussian noise, *Appl. Math. Comput.*, **429** (2022), 127218. <https://doi.org/10.1016/j.amc.2022.127218>
33. M. Kamrani, N. Jamshidi, Implicit euler approximation of stochastic evolution equations with fractional Brownian motion, *Commun. Nonlinear Sci. Numer. Simul.*, **44** (2017), 1–10. <https://doi.org/10.1016/j.cnsns.2016.07.023>
34. L. Xu, Z. Li, J. Luo, Global attracting set and exponential decay of second-order neutral stochastic functional differential equations driven by fBm, *Adv. Differ. Equations*, **134** (2017). <https://doi.org/10.1186/s13662-017-1186-2>
35. G. Gripenberg, I. Norros, On the prediction of fractional Brownian motion, *J. Appl. Probab.*, **33** (1996), 400–410. <https://doi.org/10.2307/3215063>
36. Y. Hu, X. Y. Zhou, Stochastic control for linear systems driven by fractional noises, *SIAM J. Control Optim.*, **43** (2005), 2245–2277. <https://doi.org/10.1137/S0363012903426045>

37. L. Lu, X. Meng, Z. Mao, G. E. Karniadakis, DeepXDE: A deep learning library for solving differential equations, *SIAM Rev.*, **63** (2021), 208–228. <https://doi.org/10.1137/19M1274067>
38. C. Zeng, Y. Chen, Q. Yang, The fBm-driven Ornstein-Uhlenbeck process: Probability density function and anomalous diffusion, *Fract. Calc. Appl. Anal.*, **15** (2012), 479–492. <https://doi.org/10.2478/s13540-012-0034-z>
39. A. V. Mikhajlovich, V. V. Ivanovich, Analysis of statistical characteristics of probability density distribution of the signal mixture and additive-multiplicative non-gaussian noise, in *2019 Dynamics of Systems, Mechanisms and Machines (Dynamics)*, (2019), 1–6. <https://doi.org/10.1109/Dynamics47113.2019.8944670>
40. S. C. Q. Valente, R. D. C. L. Bruni, Z. G. Arenas, D. G. Barci, Effects of multiplicative noise in bistable dynamical systems, *Entropy*, **27** (2025), 155. <https://doi.org/10.3390/e27020155>
41. G. Volpe, J. Wehr, Effective drifts in dynamical systems with multiplicative noise: A review of recent progress, *Rep. Prog. Phys.*, **79** (2016), 053901. <https://doi.org/10.1088/0034-4885/79/5/053901>



AIMS Press

© 2025 the Author(s), licensee AIMS Press. This is an open access article distributed under the terms of the Creative Commons Attribution License (<https://creativecommons.org/licenses/by/4.0>)