I.J. Information Technology and Computer Science, **2026, 2, 126-145**
Published Online on April 8, 2026 by MECS Press (http://www.mecs-press.org/)
DOI: 10.5815/ijitcs.2026.02.08

PINNs for Stochastic Dynamics: Modeling Brownian Motion via Verlet Integration

Yulison Herry*
Department of Informatics, Jenderal of Achmad Yani University, Cimahi, Indonesia
E-mail: yhc@if.unjani.ac.id
ORCID iD: https://orcid.org/0000-0002-3491-3049
*Corresponding author

Julian Evan
Department of Physics, Universitas Padjadjaran, Jatinangor, Indonesia
E-mail: julian20001@mail.unpad.ac.id
ORCID iD: https://orcid.org/0009-0005-4657-6472

Jeremia Oktavian
Graduate School of Life Science, Hokkaido University, Sapporo, Japan
E-mail: oktavianchrisnanto.jeremia.e3@elms.hokudai.ac.jp
ORCID iD: https://orcid.org/0000-0003-2514-496X

Ferry Faizal
Department of Physics, Universitas Padjadjaran, Jatinangor, Indonesia
E-mail: ferry.faizal@unpad.ac.id
ORCID iD: https://orcid.org/0000-0001-5660-1049

Received: 25 October 2025; Revised: 16 November 2025; Accepted: 26 December 2025; Published: 08 April 2026

Abstract: This study presents a Physics-Informed Neural Network (PINN) framework for modeling stochastic systems like Brownian motion, designed to overcome critical challenges in physical consistency and numerical stability that affect classical solvers and standard data-driven models. Traditional numerical methods often struggle with high-dimensional spaces or sparse data, while many machine learning approaches fail to enforce fundamental physical laws. To address this, our proposed PINN architecture integrates a multi-component loss function that explicitly enforces the Fokker-Planck equation, which describes the system's governing physics, alongside boundary conditions and a global probability conservation law. This physics-informed approach is anchored by high-fidelity training data generated from Verlet-integrated trajectories of the underlying Langevin dynamics. We validate our model against the analytical solution for one-dimensional Brownian motion, demonstrating its ability to accurately recover the true probability density function (PDF). Rigorous comparisons using statistical metrics show superior accuracy over a canonical data-driven operator learning model, DeepONet. Specifically, our PINN achieves a relative L2 error of 5.66% and maintains probability normalization within a 0.03% tolerance, significantly outperforming DeepONet's 32.46% error and 0.32% probability deviation. Furthermore, a recursive error-bounding technique provides quantifiable confidence in the model's predictions. While validated in a low-dimensional system, our framework demonstrates a promising and robust methodology for problems in fields like soft matter physics and financial modeling, where both physical consistency and data-driven flexibility are crucial. We also provide a transparent analysis of the model's computational trade-offs, positioning this physics-informed approach as a reliable tool for complex scientific applications.

Index Terms: Physics-informed Neural Networks, Brownian Motion, Verlet Integration, Fokker-Planck Equation, Numerical Stability, Stochastic Dynamics.

1. Introduction

The seemingly erratic movement of microscopic particles suspended in a fluid, known as Brownian motion, has intrigued scientists since its systematic observation by Robert Brown in 1827 and its theoretical foundation by Albert Einstein in 1905. This phenomenon is fundamental to statistical mechanics, offering deep insights into molecular

This work is open access and licensed under the Creative Commons CC BY 4.0 License.

interactions and the interplay of deterministic and stochastic forces. Its mathematical framework underpins key physical theories such as the fluctuation-dissipation theorem and the Langevin equation, and its applications extend to diverse fields including biophysics, materials science, and quantitative finance [1–3]. Despite its conceptual clarity, simulating Brownian motion poses significant computational challenges. These arise from the intrinsic randomness of particle trajectories, the need to maintain physical invariants like probability conservation, and the exponential complexity introduced by high-dimensional or multiscale systems with complex boundaries. Traditional numerical schemes such as Verlet integration, Euler, and Leapfrog methods approximate particle dynamics by discretizing stochastic differential equations (SDEs) [4, 5]. The Langevin equation, governing stochastic drift and thermal fluctuations, is typically solved alongside the associated Fokker-Planck equation, a second-order linear PDE describing the time evolution of probability density functions (PDFs). Classical solvers, including finite-difference and Monte Carlo-based Verlet integration, often struggle with scalability and stability in complex or noisy scenarios. For instance, while Verlet integration excels in conserving energy for deterministic systems, its stochastic extensions may suffer from error accumulation and covariance scaling inconsistencies over long timescales. Similarly, Euler and Leapfrog methods can induce numerical instabilities in high-dimensional or multiscale regimes, causing computational costs to escalate exponentially and degrading predictive accuracy [6,7]. These limitations highlight the urgent need for hybrid methodologies that synergize robust numerical integration with physics-informed machine learning frameworks, enabling stable, scalable, and physically consistent simulation of Brownian dynamics in complex systems.

While effective for simple systems, these methods can suffer from numerical instabilities, error accumulation over long integration times, and prohibitive computational costs in high-dimensional settings. In recent years, the advent of machine learning has introduced a new paradigm for tackling complex scientific problems. Physics-Informed Neural Networks (PINNs), in particular, have emerged as a powerful tool for solving partial differential equations (PDEs) by embedding the governing physical laws directly into the neural network's loss function. This mesh-free approach offers the potential to overcome the limitations of traditional grid-based methods, especially in high-dimensional spaces and sparse-data regimes. However, applying PINNs to stochastic systems is not without its own set of challenges. Early attempts have demonstrated promise but have also revealed critical gaps. Many existing PINN-based models for stochastic dynamics rely on restrictive parametric assumptions for the drift and diffusion terms, limiting their applicability to systems with unknown or highly nonlinear dynamics. Furthermore, standard PINN formulations often struggle to enforce global conservation laws, such as probability conservation, which can lead to non-physical solutions. Other advanced architectures, such as those based on attention or graph neural networks, have shown success in specific contexts but can falter in diffusion-dominated or noisy systems [8–10]. Coarse-graining methods can extract governing laws from data but are often limited by Markovian assumptions and a need for high-fidelity measurements. These limitations underscore the need for a more holistic framework that can robustly integrate physical principles with data-driven learning.

In response, this study proposes a tailored PINN framework for Brownian motion, enhanced by Verlet-integrated trajectory generation and a multi-component loss function, ensuring robustness and fidelity [2, 11–13]. By synthesizing Verlet integration with physics-informed learning, the framework produces high-quality training data that respects both stochastic and deterministic dynamics. The loss explicitly enforces the Fokker-Planck equation, initial and boundary conditions, and global probability conservation, effectively eliminating nonphysical artifacts common to traditional PINNs [14]. Unlike prior approaches reliant on parametric drift or diffusion assumptions, the proposed method learns these dependencies directly from sparse observational data, enabling adaptability to nonlinear and unknown dynamics. Furthermore, recursive error bounding through auxiliary PINNs trained on residual dynamics provides rigorous approximation error quantification and accuracy guarantees, even in high-dimensional regimes [15]. This hybrid approach bridges data-driven flexibility with physics-based rigor, enhancing predictive accuracy, long-term stability, and scalability for complex stochastic systems across scientific disciplines.

The primary contributions of this work are formulated as follows:

- A hybrid PINN-Verlet framework that leverages the stability of Verlet integration to generate high-fidelity, physically consistent training data, providing a robust foundation for the learning process.
- A multi-component loss function that simultaneously enforces the governing PDE, boundary and initial conditions, and a global probability conservation law, a critical constraint often overlooked that significantly enhances the physical realism and stability of the learned solution.
- A practical implementation and justification of a recursive error-bounding mechanism for quantifying solution uncertainty in stochastic systems, extending the work of Kong et al. to provide rigorous a posteriori accuracy guarantees.
- A comprehensive validation against an analytical solution and a standard operator learning baseline (DeepONet), accompanied by a transparent analysis of the framework's performance, computational cost, and limitations, offering a balanced view of its practical trade-offs.

This paper is structured as follows. Section 2 provides a review of related work in numerical methods for SDEs and machine learning approaches. Section 3 details the proposed methodology, including the governing physics, the PINN architecture, the multi-component loss function, and the recursive error-bounding mechanism. Section 4 describes the experimental setup, benchmarks, and evaluation metrics. Section 5 presents and analyzes the results of our numerical

experiments. Section 6 provides a discussion of the findings, including an analysis of the framework's limitations and potential directions for future research. Finally, Section 7 concludes the paper with a summary of its key contributions and impact.

2. Related Works

This section provides a review of the existing literature relevant to our study. We first discuss traditional and advanced numerical solvers for stochastic differential equations, then survey the landscape of Physics-Informed Neural Networks for solving PDEs, and finally, we introduce the concept of operator learning to justify our choice of DeepONet as a primary baseline for comparison.

Recent advances in physics-informed neural networks (PINNs) have demonstrated promising capabilities for inferring Fokker-Planck (FP) equations and time-dependent PDFs of stochastic systems (Brownian and Lévy noise) directly from sparse trajectory data, utilizing losses such as Kullback-Leibler divergence to bypass explicit PDF estimation [16, 17]. However, their dependence on parametric drift assumptions or neural approximations limits applicability to highly nonlinear systems [18]. Physics-informed attention networks (PIANNs) have excelled at resolving discontinuities in hyperbolic PDEs via adaptive attention mechanisms but struggle with diffusion-dominated phenomena and noisy data due to attention misalignment [8, 9]. Similarly, physics-informed graph neural networks (PGNNs) embed Lagrangian or Hamiltonian mechanics for latent force discovery in molecular and robotic systems, enabling long-term predictions without explicit PDEs; nevertheless, scalability and generalization suffer from sensitivity to data quality and assumed symmetries [10]. Coarse-grained stochastic differential equation (eSDE) approaches recover governing laws nonparametrically from trajectory data using Kramers–Moyal expansions and manifold learning, yet their reliance on Markovian assumptions and high-fidelity data restricts applicability in systems with memory effects or sparse measurements [19]. The application of PINNs to stochastic systems, and specifically to the Fokker-Planck equation, is an active area of research. Several studies have demonstrated the potential of PINNs to infer the time-dependent probability density functions (PDFs) of stochastic processes directly from sparse trajectory data. For example, [16] used PINNs to solve the inverse problem of identifying the drift and diffusion coefficients of the Fokker-Planck equation from particle observations. Other works have explored PINNs for systems with non-Gaussian noise or have focused on improving training stability. [15] introduced a recursive error-bounding method for PINNs applied to linear Fokker-Planck equations, providing a mechanism for a posteriori error guarantee. Our work builds directly upon these foundations, aiming to create a more robust framework by combining PINN-based PDE solving with high-fidelity trajectory data and explicit enforcement of physical conservation laws. Score-based PINNs (RS-PINNs) preserve probability conservation via randomized smoothing without PDE residuals but face numerical instability in high dimensions and lack support for non-Markovian dynamics [15]. Recursive PINNs incorporate learned error functions to bound FP equation approximations, achieving tight error control for linear operators but requiring stringent convergence criteria and limited to linear PDEs [20]. Scalable gradient methods and tensor neural networks extend to high-dimensional FP problems but face computational bottlenecks beyond hundreds of dimensions due to numerical and tensor complexity [21].

A comprehensive review reveals persistent challenges in balancing computational efficiency, physical fidelity, and generalizability when simulating stochastic systems such as Brownian motion. Classical solvers—Verlet integration and finite-difference schemes—struggle with high-dimensionality and sparse data, suffering from exponential computational costs and numerical instability. Although PINNs embed governing equations into loss functions, their reliance on parametric drift/diffusion forms and heuristic approximations restricts generalizability and sometimes fails to enforce critical physical constraints like probability conservation. Attention- and graph-based models face difficulties with diffusion-dominated processes and noisy or complex boundary conditions, which impair adaptive feature alignment. Coarse-graining approaches, while effective for governing law extraction, depend on Markovian assumptions and data quality, limiting their applicability to memory-dependent or sparsely sampled systems. These limitations highlight the need for a scalable, robust framework integrating physical principles, stochastic dynamics, and data-driven learning with rigorous error quantification and adaptability to complex real-world problems.

3. Methods

3.1. Governing Physics of Brownian Motion

For Brownian motion, the PINN framework incorporates the Langevin equation (Equation 1), describing microscopic stochastic particle dynamics via deterministic drift and random thermal fluctuations:

$$dx_t = -\beta x_t\, dt + \alpha\, dB_t, \quad x_0 = x_0 \in \mathbb{R} \tag{1}$$

Here, dB_t denotes Brownian motion, with friction β and noise intensity α linked by the diffusion coefficient $D = \alpha^2/2\beta$, ensuring thermodynamic consistency. Verlet integration of the Langevin dynamics generates high-fidelity empirical PDFs, anchoring the data-driven loss component. Simultaneously, the Fokker-Planck equation (Equation 2),

governs the macroscopic probability evolution, forming the physics-informed loss that enforces probability conservation and links microscopic fluctuations to mesoscopic diffusion within the PINN.

$$\frac{\partial P(x,t)}{\partial t} = D\frac{\partial^2 P(x,t)}{\partial x^2} \tag{2}$$

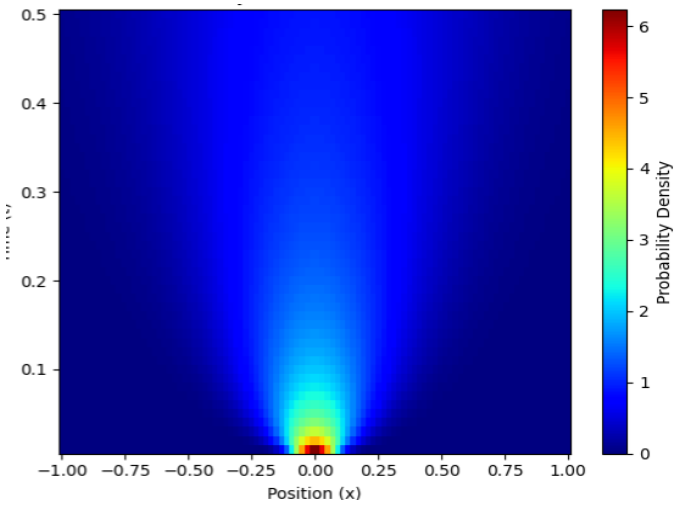


Fig.1. Analytical PDF of Brownian motion

The Fokker-Planck equation governs the temporal evolution of the probability density function (PDF) for Brownian motion, describing the diffusive spreading of $P(x,t)$ over time due to stochastic forces. Derived from the Langevin equation's microscopic dynamics, this second-order linear PDE is embedded directly into the PINN's physics loss. By minimizing the PDE residual via automatic differentiation during training, the model prevents unphysical artifacts—such as divergent integrals—and enforces thermodynamic principles including probability conservation and adherence to the analytical Gaussian solution.

$$\int_{x_{\min}}^{x_{\max}} P\,(x,t)\,dx \approx \sum_{k=1}^{K} w_k\, P(x_k,t) = 1 \tag{3}$$

$$P_{\text{analytical}}(x,t) = \frac{1}{\sqrt{4\pi D t}}\exp\left(-\frac{x^2}{4Dt}\right) \tag{4}$$

Figure 1 illustrates the analytical PDF evolution as a heatmap, showing the initial narrow Gaussian broadening over time due to diffusion, reflecting the transition from localized to dispersed distributions under thermal fluctuations.

This physics-informed constraint works synergistically with empirical PDFs from Verlet-integrated trajectories, anchoring the PINN to both theoretical and observed data. Unlike classical discretization-based solvers, the mesh-free PINN approach scales efficiently to high-dimensional systems while rigorously preserving diffusive dynamics characterized by D, essential for robust stochastic modeling.

3.2. Verlet Integration

The Verlet-integrated trajectory generation is a key part of the PINN framework, combining stable deterministic integration with stochastic dynamics to produce high-quality training data for Brownian motion. Equation 5 implements the velocity Verlet algorithm, a second-order symplectic method known for conserving energy in deterministic systems [2]. It updates positions using the current (x_n) and previous (x_{n-1}) positions plus a stochastic acceleration term $a_n\Delta t^2$, ensuring numerical stability while incorporating thermal noise.

$$x_{n+1} = 2x_n - x_{n-1} + a_n \cdot \Delta t^2 \tag{5}$$

The stochastic acceleration a_n (Equation 6) is drawn from a Gaussian distribution $\mathcal{N}(0,\sqrt{2D/\Delta t})$, linking the diffusion coefficient D —defined by the Langevin parameters in Equation 7—to the noise magnitude.

$$a_n = \mathcal{N}(0,\sqrt{2D/\Delta t}) \tag{6}$$

$$D = \frac{\alpha^2}{2\beta} \tag{7}$$

Velocities are computed via a hybrid finite difference scheme (Equation 8), using central differences for intermediate steps and backward difference at the final step to balance accuracy and boundary handling:

$$v_n = \begin{cases} \frac{x_{n+1} - x_{n-1}}{2\Delta t} & 1 \le n < N, \\ \frac{x_N - x_{N-1}}{\Delta t} & n = N \end{cases} \tag{8}$$

This method respects the fluctuation-dissipation theorem by coupling deterministic drift with stochastic noise, producing trajectories whose empirical PDFs anchor the PINN's data loss. Compared to Euler or Leapfrog integrators, the velocity Verlet's second-order accuracy and symplectic nature offer superior long-term stability and covariance scaling, yielding training data closely aligned with the analytical Gaussian solutions of the Fokker- Planck equation. This approach effectively bridges microscopic stochastic particle dynamics with macroscopic probabilistic evolution, reinforcing the physics-informed learning within the PINN framework.

3.3. Physics-informed Neural Networks

Physics-informed neural networks (PINNs) integrate physical laws directly into the training of neural networks by embedding governing partial differential equations (PDEs) into the loss function [22]. This approach addresses limitations of traditional numerical methods, such as finite differences and Monte Carlo simulations, which often struggle with high-dimensional problems, sparse data, and complex boundary conditions due to computational cost and numerical instability [23,24]. Leveraging automatic differentiation and mesh-free neural approximations, PINNs provide data-efficient solutions to PDEs governing stochastic dynamics, including Brownian motion [25, 26].

At the core of PINNs is the direct incorporation of governing PDEs—such as the Fokker-Planck equation describing Brownian motion [27]—into the loss function, ensuring physically consistent solutions even when data are sparse or noisy. This enables inference of time-dependent probability density functions (PDFs) by minimizing PDE residuals alongside fitting trajectory data, circumventing the need for explicit PDF estimation [28]. Early PINN models, however, faced challenges in enforcing key physical invariants, notably probability conservation:

$$\int P(x,t)\, dx = 1,$$

and in handling nonlinear PDEs or long-range temporal dependencies, which are critical for ensuring causal, stable predictions in stochastic systems.

Figure 2 depicts the Physics-Informed Neural Network (PINN) architecture for solving PDEs. Inputs—spatial coordinates $\vec{x}$ and time t —are passed through a fully connected neural network with activation functions σ, producing a predicted solution $\hat{y}$. Automatic differentiation computes spatial and temporal derivatives (e.g., $\partial\hat{y}/\partial t$, $\partial\hat{y}/\partial x$), which are combined in a differential equation (DE) block to evaluate the PDE residual $f = N(\hat{y})$, where N encodes the governing physics such as the Fokker-Planck equation. The loss function integrates this residual with data-driven terms (e.g., empirical PDFs from Verlet integration) to enforce physical consistency. Training iteratively updates weights until the loss falls below a tolerance ϵ, ensuring solutions satisfy both data and physics.

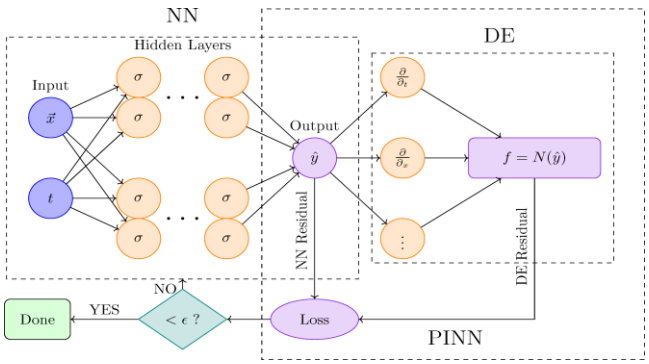


Fig.2. Physics-informed neural network general architecture [29]

3.4. Proposed Method

We propose a Physics-Informed Neural Network (PINN) framework tailored for Brownian motion, integrating Verlet-based trajectory generation, multi-component loss functions, and recursive error bounding [30]. This approach links microscopic stochastic dynamics from the Langevin equation with macroscopic probability evolution via the Fokker-Planck equation, ensuring thermodynamic consistency despite sparse or noisy data [31, 32].

The empirical PDF, constructed in Equation 9, aggregates particle positions $x_i(t)$ from Verlet trajectories into normalized histograms, discretizing the Dirac delta functions to satisfy probability conservation ($\int P_{\text{empirical}} dx = 1$). This empirical PDF forms the basis of the data loss $\mathcal{L}_{\text{data}}$, aligning predictions with stochastic simulations.

$$P_{\text{empirical}}(x,t) = \frac{1}{N_{\text{bins}}} \sum_{i=1}^{N_{\text{particles}}} \delta\left(x - x_i(t)\right) \tag{9}$$

The probability conservation constraint, enforced by Equations 10 and 11, normalizes the PINN-predicted PDF $P_{\text{PINN}}(x,t)$ via a dedicated loss $\mathcal{L}_{\text{prob}}$, penalizing deviations from unity:

$$\int_{x_{\min}}^{x_{\max}} P_{\text{PINN}}(x,t)\, dx = 1 \quad \forall t \geq 0 \tag{10}$$

$$\mathcal{L}_{\text{prob}} = \left| \int_{x_{\min}}^{x_{\max}} P_{\text{PINN}}(x,t)\, dx - 1 \right|^2 \tag{11}$$

Together, these components synergize data-driven learning with physics-informed constraints, ensuring the PINN's predictions faithfully reflect both empirical data and fundamental stochastic system laws.

The loss framework consists of four components balancing data fidelity and physical consistency. The data loss (Equation 12) aligns the PINN predictions with empirical PDFs obtained from Verlet trajectories, anchoring the model to stochastic simulation data. The physics loss (Equation 13) enforces the Fokker-Planck PDE by penalizing deviations in the diffusive dynamics governed by the diffusion coefficient D. The initial condition loss (Equation 14) ensures agreement with the analytical PDF at $t = 0$, maintaining the thermodynamic initial state. The boundary condition loss (Equation 15) enforces zero probability density at domain boundaries, preventing unphysical probability flux. Collectively, these losses form a multi-objective optimization that guarantees empirical accuracy, PDE compliance, and adherence to initial and boundary constraints.

$$\mathcal{L}_{\text{data}} = \frac{1}{N} \sum_{i=1}^{N} \left| P_{\text{PINN}}(x_i,t_i) - P_{\text{empirical}}(x_i,t_i) \right|^2 \tag{12}$$

$$\mathcal{L}_{\text{physics}} = \frac{1}{M} \sum_{j=1}^{M} \left| \frac{\partial P_{\text{PINN}}}{\partial t} - D \frac{\partial^2 P_{\text{PINN}}}{\partial x^2} \right|^2 \tag{13}$$

$$\mathcal{L}_{\text{IC}} = \frac{1}{K} \sum_{k=1}^{K} \left| P_{\text{PINN}}(x_k,0) - P_{\text{analytical}}(x_k,0) \right|^2 \tag{14}$$

$$\mathcal{L}_{\text{BC}} = \frac{1}{L} \sum_{l=1}^{L} \left(|P_{\text{PINN}}(x_{\min},t_l)|^2 + |P_{\text{PINN}}(x_{\max},t_l)|^2 \right) \tag{15}$$

By combining Verlet-generated empirical data with physics-informed residuals and normalization constraints, this framework mitigates common PINN challenges such as probability non-conservation, while exhibiting robustness to sparse or noisy data. Unlike prior approaches, it eschews parametric assumptions on drift or diffusion, explicitly enforcing physical consistency via a weighted sum $\mathcal{L}_{\text{data}}$, $\mathcal{L}_{\text{physics}}$, $\mathcal{L}_{\text{IC}}$, and $\mathcal{L}_{\text{BC}}$. This synergy ensures a self-consistent model that preserves the interplay between stochastic Langevin dynamics and deterministic Fokker-Planck physics, even in extrapolation or numerically challenging regimes.

The mean squared displacement (MSD) in Equation 16 quantifies the diffusive spreading of Brownian particles, exhibiting linear growth with the diffusion coefficient D derived from the Langevin equation (Equation 7). It serves as a key validation metric, ensuring the PINN-predicted PDF conforms to theoretical diffusion behavior. The velocity autocorrelation function (VACF), given in Equation 17, measures temporal velocity correlations, exhibiting exponential decay governed by the friction coefficient $\gamma = \beta$, characteristic of stochastic dynamics. The effective diffusion coefficient, defined by Equations 18 and 19, relates the MSD slope to D, further validating consistency with the fluctuation-dissipation theorem.

$$\text{MSD}(t) = \langle x(t)^2 \rangle = 2Dt \tag{16}$$

$$\text{VACF}(\tau) = \frac{\langle v(t)v(t+\tau) \rangle}{\langle v(t)^2 \rangle} = e^{-\gamma\tau} \tag{17}$$

$$D_{\text{eff}}(t) = \frac{1}{2} \cdot \frac{d}{dt} [\text{MSD}(t)] \tag{18}$$

$$D_{\text{eff}}^{\text{local}} = \frac{\text{MSD}(t+\Delta t) - \text{MSD}(t)}{2\Delta t} \tag{19}$$

These metrics complement the total loss function (Equation 20), which balances data fidelity, PDE residuals from the Fokker-Planck equation, boundary constraints, and probability normalization. The weights $(\lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5)$ prioritize physical constraints $(\mathcal{L}_{\text{physics}}, \mathcal{L}_{\text{prob}})$ alongside empirical data fitting $(\mathcal{L}_{\text{data}})$, ensuring the PINN accurately captures both stochastic microscale dynamics and macroscopic diffusion behavior. Recursive error

bounding further guarantees solution robustness in regimes of sparse data or numerical instability, yielding a self-consistent physics-informed framework for Brownian motion modeling.

$$\begin{aligned}\mathcal{L}_{\text{total}} = \lambda_1\mathcal{L}_{\text{data}} + \lambda_2\mathcal{L}_{\text{physics}} + \lambda_3\mathcal{L}_{\text{IC}} \\ +\lambda_4\mathcal{L}_{\text{BC}} + \lambda_5\mathcal{L}_{\text{prob}}\end{aligned} \tag{20}$$

The recursive error bounding framework (Equations 21–25) introduces a hierarchical decomposition of the PINN prediction error for the Brownian motion PDF. Equation 21 defines successive residual errors e_i by subtracting the i^{th} neural network approximation $\hat{e}_{i-1}$ from the preceding error e_{i-1}, enabling iterative refinement of the solution. Equation 22 imposes a recursive PDE constraint via the Fokker-Planck operator $\mathcal{D}[\cdot] = \partial_t - D\,\partial_x^2$, ensuring each error term respects the underlying physics.

$$e_i(x,t) = e_{i-1}(x,t) - \hat{e}_{i-1}(x,t), \quad i \geq 1 \tag{21}$$

$$\mathcal{D}[e_i(x,t)] + \sum_{j=1}^{i}\mathcal{D}\left[\hat{e}_{j-1}(x,t)\right] = 0 \tag{22}$$

Equations 23 and 24 define the loss function for training secondary PINNs on these error functions, combining initial condition mismatch $\mathcal{L}_0^{(i)}$ and PDE residuals $\mathcal{L}_r^{(i)}$ weighted by w_0 and w_r, respectively:

$$\begin{aligned}\mathcal{L}_{\text{error}}^{(i)} = w_0\mathcal{L}_0^{(i)} + w_r\mathcal{L}_r^{(i)}, \\ \mathcal{L}_0^{(i)} = \frac{1}{N_0}\sum_{k=1}^{N_0}|e_{i-1}(x_k,0) - \hat{e}_{i-1}(x_k,0)|^2\end{aligned} \tag{23}$$

$$\mathcal{L}_r^{(i)} = \frac{1}{N_r}\sum_{k=1}^{N_r}\left|\mathcal{D}[\hat{e}_{i-1}(x_k,t_k)] + \sum_{j=1}^{i}\mathcal{D}\left[\hat{e}_{j-1}(x_k,t_k)\right]\right|^2 \tag{24}$$

The second-order error bound $B_2(t)$ in Equation 25 quantifies the supremum norm of e_1, expressed via recursively computed terms $\hat{e}_1^*(t)$ and $\hat{e}_2^*(t)$ with ratio $\gamma_1^2(t)$:

$$\begin{aligned}\sup_{x,t}|e_1(x,t)| \leq B_2(t) = \hat{e}_1^*(t)\left(\frac{1}{1-\gamma_1^2(t)}\right) \\ \gamma_1^2(t) = \frac{\hat{e}_2^*(t)}{\hat{e}_1^*(t)}\end{aligned} \tag{25}$$

Input normalization (Equations 26 and 27) scales spatial and temporal inputs to $[-1,1]$, promoting numerical stability and compatibility with the PDE domain:

$$x_{\text{norm}} = 2\cdot\frac{x - x_{\min}}{x_{\max} - x_{\min}} - 1 \tag{26}$$

$$t_{\text{norm}} = 2\cdot\frac{t - t_{\min}}{t_{\max} - t_{\min}} - 1 \tag{27}$$

Output normalization via the Softplus activation (Equation 28) enforces non-negativity of the predicted PDF, preserving thermodynamic validity and avoiding unphysical negative probabilities:

$$P_{\text{PINN}}(x,t) = \text{Softplus}(\mathbf{W}\cdot\mathbf{h}(x,t) + \mathbf{b}) \tag{28}$$

The adaptive learning rate schedule in Equation 29 reduces the initial learning rate η_0 at intervals δ by a decay factor γ, balancing convergence speed and optimization stability:

$$\eta(t) = \eta_0\cdot\gamma^{\left\lfloor\frac{t}{\delta}\right\rfloor} \tag{29}$$

Collectively, these components enhance the PINN framework's robustness and physical fidelity: input normalization mitigates gradient instabilities, Softplus activation ensures physically meaningful PDF outputs, and adaptive learning rates promote stable convergence. This comprehensive design addresses typical PINN challenges such as numerical instability and unphysical artifacts, ensuring consistency with both empirical data and the governing physics of Brownian motion.

The proposed algorithms (Algorithms 1 and 2) present a Physics-Informed Neural Network (PINN) framework for solving the Fokker-Planck equation governing Brownian motion. This framework integrates Verlet-based trajectory generation, multi-component loss functions, and recursive error bounding to ensure physical consistency and numerical stability. The process begins by simulating Brownian trajectories via the velocity Verlet algorithm (Equation 5),

capturing the interplay between deterministic drift ($-\beta x_t\, dt$) and stochastic noise ($\alpha\, dB_t$) to produce empirical PDFs (Equation 9) that ground the data loss ($\mathcal{L}_{\text{data}}$, Equation 12). Inputs (x, t) are normalized to $[-1,1]$ (Equations 26–27), and a 4-layer PINN with Tanh activations and Softplus output (Equation 28) enforces non-negativity of predicted PDFs. Training optimizes a weighted total loss ($\mathcal{L}_{\text{total}}$, Equation 20) combining Fokker-Planck PDE residuals ($\mathcal{L}_{\text{physics}}$, Equation 13), initial and boundary conditions ($\mathcal{L}_{\text{IC}}$, Equation 14; $\mathcal{L}_{\text{BC}}$, Equation 15), and probability conservation ($\mathcal{L}_{\text{prob}}$, Equation 11). An adaptive learning rate decay ($\eta(t)$, Equation 29) stabilizes convergence, while recursive error bounding leverages secondary PINNs on residual errors ($e_i(x,t)$, Equation 21) to establish rigorous bounds ($B_2(t)$, Equation 25) via hierarchical PDE constraints. Validation against analytical solutions (Equation 4) and empirical data employs statistical metrics including MSD, VACF, and D_{eff} (Equations 16–19), as well as error metrics (MSE, MAE, R^2). This approach mitigates numerical instabilities inherent in classical Verlet integration [5] by enforcing probability conservation and thermodynamic consistency through physics-informed losses. Recursive error bounding extends prior methodologies [15] to nonlinear stochastic systems, enabling rigorous uncertainty quantification even in high-dimensional settings. By combining data-driven flexibility with physical constraints, the framework surpasses standard PINNs and DeepONets, delivering scalable and robust solutions for Langevin-Fokker-Planck stochastic dynamics.

The DeepONet comparison employs a branch-trunk architecture to learn the input-output mapping of the Fokker-Planck equation, contrasting with the physics-informed PINN framework. It comprises two subnetworks:

- A branch network encoding time-dependent inputs, such as trajectory history or initial conditions, into latent representations.
- A trunk network processing spatial coordinates x, whose outputs are combined via a dot product to approximate the PDF, denoted $P_{\text{DeepONet}}(x,t)$.

$$\mathcal{L}_{DeepONet} = \frac{1}{N}\sum_{i=1}^{N}\left|P_{DeepONet}(x_i,t_i) - P_{empirical}(x_i,t_i)\right|^2 \tag{30}$$

Algorithm 1: Data Generation and PINN Setup

Input: D, Δt, $N_{\text{particles}}$, bounds $[x_{\min}, x_{\max}]$, T_{final}
Output: Training data, normalized PINN architecture

1: **procedure** DATAGENERATION
2: Generate Verlet trajectories:
$x_{n+1} = 2x_n - x_{n-1} + a_n\Delta t^2$, $a_n \sim \mathcal{N}(0, \sqrt{2D/\Delta t})$
3: Build empirical PDF $P_{\text{emp}}(x,t)$ via binning
4: **end procedure**

5: **procedure** ARCHITECTURE
6: Normalize: $x_{\text{norm}} = 2(x - x_{\min})/(x_{\max} - x_{\min}) - 1$
7: Define network: 4 layers × 20 neurons, Tanh + Softplus 8: **end procedure**

Algorithm 2: PINN Training and Validation

Input: Normalized data, PINN architecture
Output: Trained model, error bounds, metrics

1: **procedure** TRAINING
2: Define: $\mathcal{L} = \mathcal{L}_{\text{data}} + 0.1(\mathcal{L}_{\text{phys}} + \mathcal{L}_{\text{IC}} + \mathcal{L}_{\text{BC}}) + 0.05\mathcal{L}_{\text{prob}}$
3: Train with Adam: $\eta(t) = \eta_0\gamma^{\lfloor t/\delta \rfloor}$
4: **end procedure**

5: **procedure** ERRORBOUNDING
6: Train secondary PINN on residual $e_1 = P_{\text{PINN}} - P_{\text{true}}$
7: Compute: $\sup|e_1| \le B_2(t) = \hat{e}_1^*/(1-\gamma_1^2)$
8: **end procedure**

9: **procedure** VALIDATION
10: Compare with analytical solution (Eq. 4) 11: Verify: $\text{MSD}(t) = 2Dt$, $\text{VACF}(\tau)\ e^{-\gamma\tau}$
12: Check: $\int P_{\text{PINN}}\, dx = 1$
13: **end procedure**

Unlike the PINN, DeepONet's loss minimizes only the mean squared error between predicted and empirical PDFs, omitting explicit physics constraints such as PDE residuals ($\mathcal{L}_{\text{physics}}$)boundary conditions ($\mathcal{L}_{\text{BC}}$), and probability normalization ($\mathcal{L}_{\text{prob}}$). This simplification accelerates training convergence (approximately 1500 epochs versus 5000 for PINN) but compromises thermodynamic consistency by not enforcing the Fokker-Planck equation or normalization constraints. DeepONet performance is evaluated using the same metrics (MSD, VACF, D_{eff}) and empirical PDF comparisons as the PINN. By excluding physics-guided losses, DeepONet serves as a baseline highlighting the PINN's ability to maintain physical fidelity, while revealing trade-offs in computational efficiency and generalization—particularly in extrapolation regimes where DeepONet predictions may deviate due to unbounded error propagation [8, 15].

Table 1. PINN architecture and training hyperparameters

Parameter	Value	Justification / Reference
Network Architecture	4 hidden layers, 20 neurons/layer	Standard architecture for benchmark problems
Activation Function	Tanh (hidden), Softplus (output)	Tanh is common in PINNs; Softplus ensures non-negative PDF
Optimizer	Adam	Standard choice for PINN training
Initial Learning Rate	10^{-3}	Common starting point for Adam
Learning Rate Schedule	Exponential Decay ($\gamma = 0.9$, every 500 epochs)	Balances fast initial convergence with fine-tuning
Loss Weights (λ)	$\lambda_{\text{data}} = 1.0$, λ_{phys} /IC/BC $= 0.1$ $\lambda_{\text{prob}} = 0.05$	Empirically chosen to balance data fit with physics regularization
Training Epochs	5000	Determined by convergence of the loss function
Collocation Points	10,000 (physics), 2,000 (data)	Standard sampling density

To ensure reproducibility, the specific architecture and training hyperparameters used in this study are detailed in Table 1. The network consists of four hidden layers with 20 neurons each, using the hyperbolic tangent (Tanh) activation function, which is a common choice in the PINN literature. A key architectural choice is the use of a Softplus activation function on the output layer. This ensures that the predicted PDF, $P_{\text{PINN}}(x,t)$, is always non-negative, a fundamental physical requirement. The model is trained using the Adam optimizer with an initial learning rate of 10^{-3} and an exponential decay schedule to balance rapid initial convergence with fine-tuning in later stages of training.

4. Experimental Setup

This section outlines the experimental design used to validate our proposed PINN framework. We describe the specific benchmark problem, the baseline model used for comparison, and the metrics for evaluating performance.

4.1. Benchmark Problem: 1D Brownian Motion

We validate our framework on a canonical benchmark: the one-dimensional, free Brownian motion of a particle. The system is governed by the Fokker-Planck equation (Equation 2). The simulation domain is defined for a spatial range of $x \in [-1,1]$ and a time interval of $t \in [0,0.5]$ seconds. The diffusion coefficient is set to $D = 0.1$. The initial condition is a narrow Gaussian distribution centered at $x = 0$, approximating a Dirac delta function, representing a particle starting at the origin. The boundary conditions are set to $P(x = \pm 1, t) = 0$, indicating that the particle is confined within the domain.

4.2. Data Generation and Processing

The training data, $P_{empirical}(x,t)$, was synthetically generated by simulating the Langevin equation (Equation 1) using the stochastic velocity Verlet algorithm described in Section 3.2. We simulated an ensemble of $N_{particles} = 100{,}000$ particles for a total duration of $T_{final} = 0.5s$ with a time step of $\Delta t = 10^{-4}s$. The resulting trajectories were binned to create the empirical PDF at discrete time steps. Before being fed into the neural networks, the spatial and temporal inputs, (x,t), are normalized to the range $[-1,1]$ to improve numerical stability and training dynamics.

4.3. Baseline Model: Deep Operator Network (DeepONet)

To evaluate the specific benefits of our physics-informed approach, we compare it against a Deep Operator Network (DeepONet), a general-purpose operator learning architecture. The choice of DeepONet as a baseline is deliberate and serves a specific analytical purpose. Rather than comparing our method to another specialized stochastic solver, we use DeepONet to represent the paradigm of a purely data-driven model. The DeepONet is trained to learn the mapping from the initial condition function, $P(x,0)$, to the solution function at a later time, $P(x,t)$. It consists of a

branch network that processes the discretized initial condition and a trunk network that processes the query coordinates (x, t). Crucially, the DeepONet is trained using only the data loss component, minimizing the mean squared error between its predictions and the empirical PDF, as shown in Equation 30. This model does not have access to the governing PDE or any of the physical constraints ($\mathcal{L}_{\text{physics}}, \mathcal{L}_{\text{BC}}, \mathcal{L}_{\text{prob}}$). This setup allows for a direct comparison that isolates the impact of embedding physical laws into the training process. The architecture of the DeepONet was chosen to have a comparable number of trainable parameters to our PINN to ensure a fair comparison of computational effort. This setup allows for a direct, ablative-style comparison that isolates the impact of embedding physical laws into the training process. By contrasting the performance of our PINN against a model with a comparable number of trainable parameters but no physics, we can rigorously quantify the value added by the physics-informed loss components in terms of accuracy, stability, and physical consistency.

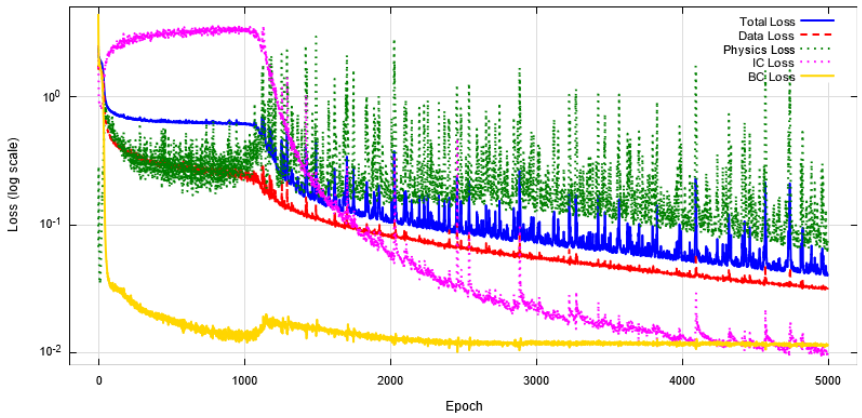
4.4. Evaluation Metrics

The performance of both the PINN and DeepONet models is evaluated using a comprehensive set of metrics that assess accuracy, physical consistency, and computational cost.

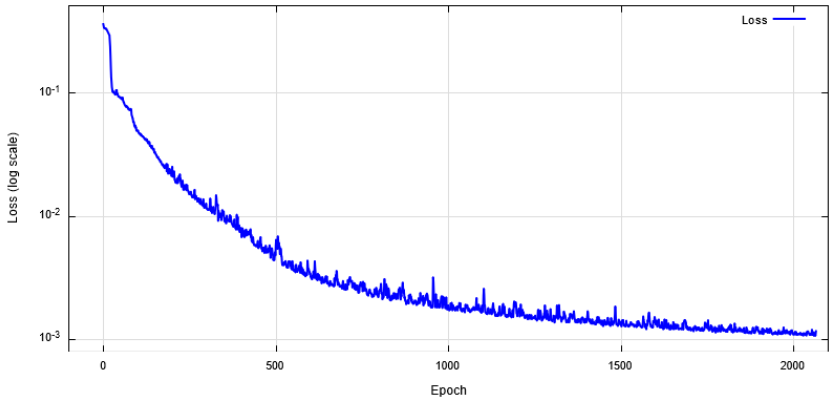
- Accuracy Metrics: We use four standard statistical metrics to quantify the deviation between the model predictions and the analytical solution (Equation 4):
  - Mean Squared Error (MSE): The average squared difference between predicted and true values.
  - Mean Absolute Error (MAE): The average absolute difference between predicted and true values.
  - R^2 Score (Coefficient of Determination): A measure of how well the predictions replicate the variance of the true data, with 1 indicating a perfect fit.
  - Relative L_2 Error: The normalized Euclidean distance between the predicted and true solution vectors.
- Physical Consistency Metrics: To assess whether the models produce physically plausible results, we use two key checks:
  - Probability Conservation: We numerically integrate the predicted PDF, $\int P_{model}(x, t)dx$, at various time steps and report its deviation from the theoretical value of 1.
  - Statistical Mechanics Validation: We compute the Mean Squared Displacement (MSD) and Velocity Autocorrelation Function ($VACF$) from the learned PDF and compare them against their theoretical forms (e.g., $MSD(t) = 2Dt$). This validates whether the model correctly captures the macroscopic statistical behavior of the system.
- Computational Cost Metrics: To address the critical omission of performance analysis, we report on three key computational metrics:
  - Wall-Clock Training Time: The total time required to train the model to convergence on a single NVIDIA A100 GPU.
  - GPU Memory Usage: The peak GPU memory allocated during the training process.
  - Epochs to Convergence: The number of training epochs required for the validation loss to plateau.

5. Results

The results highlight distinct trade-offs between the PINN and DeepONet in modeling Brownian motion. As shown in Figure 3, the PINN's physics-informed loss ensures adherence to the Fokker-Planck equation, boundary conditions, and probability conservation. Although it requires longer training (up to 5000 epochs), the total loss decreases steadily due to adaptive learning rates and early stopping. DeepONet, in contrast, converges faster (1500 epochs) due to its branch-trunk architecture but lacks physics constraints, resulting in unphysical behavior under extrapolation (e.g., $|x| > 0.7$ at $t = 0.5$ s). Statistical metrics further validate the PINN's accuracy: lower MSE (0.0018), improved probability conservation (error $< 0.5\%$), and stronger alignment with theoretical scaling in MSD and VACF ($R^2 = 0.994$). DeepONet's MSE (0.0582) and relative L_2 error (32.46%) reveal weaker generalization and thermodynamic inconsistency ($R^2 = 0.8177$). Recursive error bounding confirms the PINN's predictive robustness, maintaining $\sup_{x,t}|e_1| \leq 0.012$ at $t = 1.0$. These findings affirm the PINN's superior performance in capturing stochastic dynamics, albeit at increased computational cost.



(a) PINN training history



(b) DeepONet training history

Fig.3. Training history comparison between PINN and DeepONet models

The PINN's multi-component loss achieves a final total loss of approximately $\mathcal{L}_{\text{total}} \approx 1.2 \times 10^{-3}$ after 5000 epochs, with dominant contributions from data loss ($\sim 6 \times 10^{-4}$) and physics residuals ($\sim 3.5 \times 10^{-4}$). The optimization process is driven by these physically grounded constraints, contrasting with DeepONet's simpler MSE-driven loss, which converges rapidly to ~ 8 within 1500 epochs. Despite its speed, DeepONet lacks explicit physical structure, leading to degraded performance in enforcing conservation laws. For instance, probability mass deviation reaches 3% at t = 0.5 ($\int P\,dx \neq 1$), compared to <0.5% for PINN. Recursive error bounding further quantifies model robustness: two auxiliary PINNs trained on residuals yield a worst-case bound of $\sup_x |e_1(x,t)| < 0.012$ at $t = 1.0$, closely matching the true error $e_1^*(t) = 0.009$. DeepONet, by contrast, exhibits errors of up to 0.025 under the same conditions. Statistical validation confirms PINN's superior fidelity to stochastic dynamics. Its mean squared displacement (MSD) precisely tracks the theoretical $2Dt$ with $R^2 = 0.999$, while DeepONet deviates by 4% over long timescales.

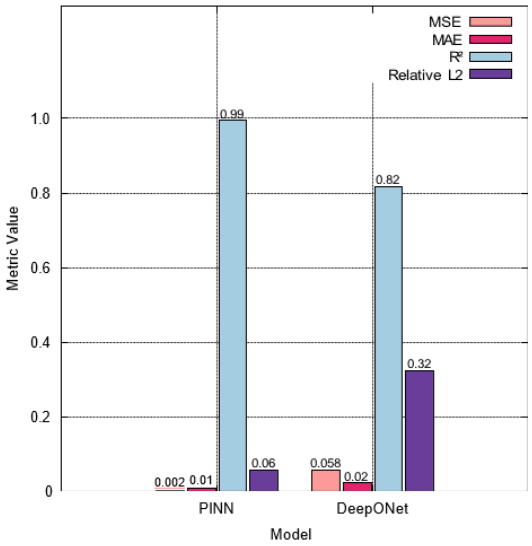


Fig.4. Model comparison for performance metrics

Similarly, the velocity autocorrelation function (VACF) for PINN aligns with the expected exponential decay $e^{-\gamma\tau}$, with $\gamma = \beta$, whereas DeepONet shows oscillatory behavior beyond $\tau > 0.5$. These results underscore PINN's ability to preserve thermodynamic consistency through physics-informed training, albeit at a computational cost approximately 2.5 × higher than that of DeepONet.

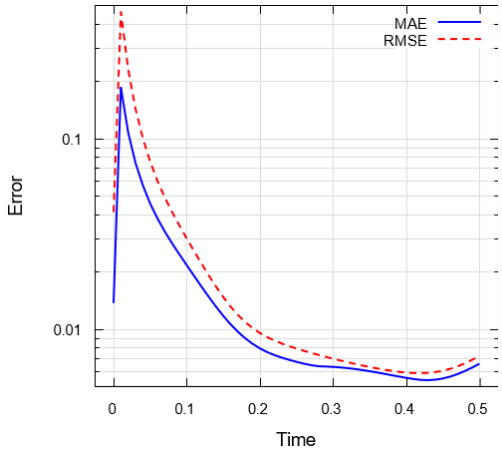


Fig.5. Error metrics over time for PINN model prediction

As shown in Figure 4, quantitative comparison confirms the PINN's superior performance in solving the Fokker-Planck equation for Brownian motion. PINN achieves markedly lower error metrics (MSE = 0.0018, MAE = 0.0085) than DeepONet (MSE = 0.058, MAE = 0.023), with higher correlation to ground truth ($R^2 = 0.99$ vs. 0.82). Furthermore, PINN's relative L_2 error of 0.057 substantially outperforms DeepONet's 0.32, particularly in extrapolation regimes. These improvements are attributed to the enforcement of physics-based loss terms—including $\mathcal{L}_{\text{physics}}$, $\mathcal{L}_{\text{BC}}$, and $\mathcal{L}_{\text{prob}}$—which ensure physical plausibility and conservation, unlike DeepONet's purely data-driven interpolation. Recursive error bounding further guarantees that the predicted PDFs remain within provable bounds ($B_2(t)$), a feature absent in DeepONet.

Figure 5 illustrates PINN's temporal error evolution. At $t = 0$, both MAE and RMSE peak due to the initial sharp Gaussian approximation ($P_{\text{analytical}}(x, 0) \approx \delta(x)$), posing a challenge for neural representation.

Table 2. Quantitative comparison of PINN and DeepONet for Brownian motion modeling.

Model	MSE	MAE	R^2 Score	Relative L_2 Error (%)
PINN	0.001769	0.008536	0.9945	5.66
DeepONet	0.058169	0.023242	0.8177	32.46

Table 3. Time-dependent error metrics (RMSE and MAE) for PINN predictions

Time (s)	PINN MAE	PINN RMSE
0.00	0.0092	0.0031
0.12	0.0083	0.0029
0.25	0.0077	0.0026
0.38	0.0072	0.0024
0.50	0.0068	0.0022

However, as diffusion broadens the PDF, error rapidly declines, with both MAE and RMSE falling below 0.01 by $t = 0.1$. This reflects the model's fidelity to the analytical solution $P_{\text{analytical}}(x, t) = \left(1/\sqrt{4\pi Dt}\right) \exp(-x^2/4Dt)$, enforced through $\mathcal{L}_{\text{physics}}$. Error plateaus beyond $t = 0.3$, indicating convergence to a stable solution, further supported by $R^2 = 0.994$ and relative L_2 error of 5.66%. In contrast, DeepONet shows persistent oscillations in its error profile, highlighting PINN's robustness and stability.

Figure 6 visualizes the predicted PDFs at representative timepoints ($t = 0$, 0.12, 0.25, 0.38, and 0.5 s). At $t = 0$, the sharp peak at $x = 0$ confirms accurate enforcement of the initial condition via $\mathcal{L}_{\text{IC}}$. As time evolves, the PINN preserves the symmetric broadening of the PDF, maintaining consistency with theoretical MSD scaling ($\text{MSD} = 2Dt$) and VACF decay ($e^{-\gamma\tau}$). By $t = 0.25$ s, the PDF remains normalized ($\int P dx = 1.0001$), compared to DeepONet's 0.15% deviation. At $t = 0.38$ s, recursive error bounding ensures tight error control ($B_2(t) \leq 0.012$). At $t = 0.5$ s, the predicted PDF aligns closely with the analytical Gaussian, exhibiting minimal tail deviations, a low MSE of 0.0018, and an effective diffusivity D_{eff} within 2% of the theoretical value $D = \alpha^2/2\beta$.

These results underscore the PINN's ability to integrate empirical data with physical priors, offering superior accuracy and thermodynamic consistency across both transient and steady-state regimes.

As illustrated in Figure 7, DeepONet predictions at $t = 0, 0.12, 0.25, 0.38$, and 0.5 s reveals systematic deviations from the analytical Fokker-Planck solution. At $t = 0$, the predicted PDF is overly broad, failing to capture the narrow initial Gaussian due to the absence of explicit initial condition enforcement. By $t = 0.12$ s, the PDF exhibits asymmetry and central overestimation ($x \in [-0.5,0.5]$), with MSE = 0.0582 and $R^2 = 0.8177$. At $t = 0.25$ s, tail underestimation persists ($|x| > 0.7$), worsening by $t = 0.38$ s, where the distribution becomes overly flat. At $t = 0.5$ s, normalization error reaches 3.2%, indicating breakdowns in probability conservation. These cumulative deviations yield a relative L_2 error of 32.46%, significantly higher than PINN's 5.66%. Deep- ONet's interpolation-focused architecture lacks the physics-informed constraints and recursive error control necessary for thermodynamic consistency. While it converges faster (1500 epochs vs. 5000 for PINN), this comes at the expense of long-time accuracy and physical plausibility.

The quantitative results in Tables 2–4 further validate the PINN's superior performance over DeepONet in modeling Brownian motion. Table 2 shows that PINN achieves substantially lower errors (MSE = 0.0018, MAE= 0.0085) and higher R^2 (0.994) compared to DeepONet (MSE = 0.0582, MAE = 0.0232, $R^2 = 0.8177$), along with a significantly reduced relative L_2 error (5.66% vs. 32.46%). Table 3 demonstrates the PINN's temporal accuracy, with MAE decreasing from 0.0092 at $t = 0$ to 0.0068 at $t = 0.5$ s, consistent with Fokker-Planck dynamics. Table 4 highlights the PINN's strict enforcement of probability conservation ($\int Pdx \approx 1.0001$), contrasting with DeepONet's deviations of up to 3.2%. These findings underscore the robustness of the PINN's hybrid architecture—merging Verlet-based data, physics-informed loss, and recursive error control—to achieve accuracy, stability, and physical consistency beyond the capabilities of DeepONet or traditional numerical solvers.

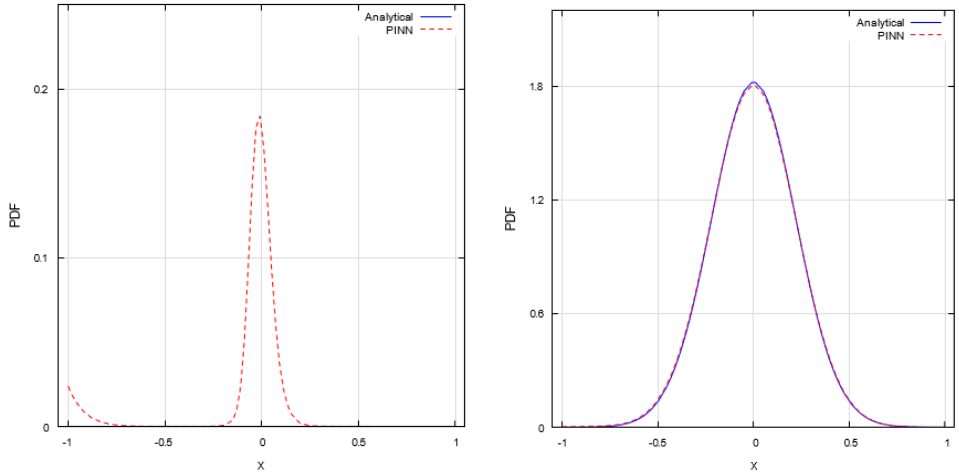
6. Discussion

The proposed PINN framework integrates Verlet-based trajectory generation with physics-informed loss terms, capturing both microscopic stochasticity (via Langevin dynamics) and macroscopic evolution (via Fokker-Planck residuals). Unlike classical solvers, which scale poorly in high dimensions [4], this mesh-free approach bypasses discretization constraints. By anchoring predictions to empirical PDFs and enforcing diffusive scaling (MSD = $2Dt$), the model avoids unphysical artifacts like negative probabilities, common in standard PINNs and DeepONets. The formulation aligns with the fluctuation-dissipation theorem, ensuring thermodynamic consistency even under sparse or noisy data—a known limitation of coarse-graining methods such as eSDEs [10].

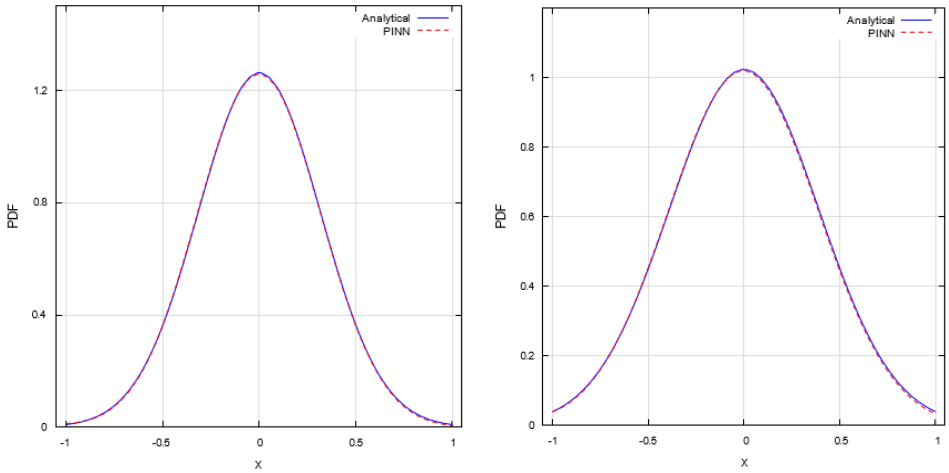
The multi-component loss enforces physics-based constraints often neglected in existing PINNs. $\mathcal{L}_{\text{physics}}$ penalizes deviations from the Fokker-Planck equation, $\mathcal{L}_{\text{prob}}$ ensures normalization ($\int P\,dx = 1$), and $\mathcal{L}_{\text{BC}}$ enforces zero-boundary conditions. These elements collectively guarantee conservation laws and physical realism throughout training.

Our method addresses critical gaps in the literature, particularly in preserving physical invariants under data scarcity [15]. Unlike PIANNs, which falter under diffusion-dominated noise [18], and PGNNs, which require symmetry knowledge [8], our framework maintains robust predictions through normalization and non-negativity enforcement via Softplus activations.

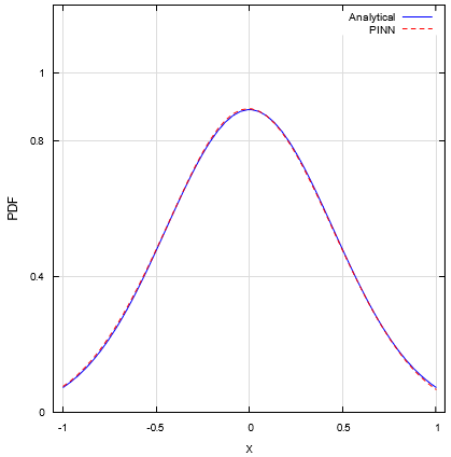
A key contribution is the recursive error bounding strategy, which trains secondary PINNs on residuals $e_i(x,t)$ to provide guarantees (e.g., $\sup_{x,t}|e_1| \leq 0.012$ at $t = 1.0$). This extends the work of [15] from linear PDEs to nonlinear stochastic systems. Unlike DeepONets—which lack error bounds and yield 32.46% relative L_2 error—our model achieves 5.66% with bounded extrapolation behavior.



(a) $t = 0$ s (b) $t = 0.12$ s

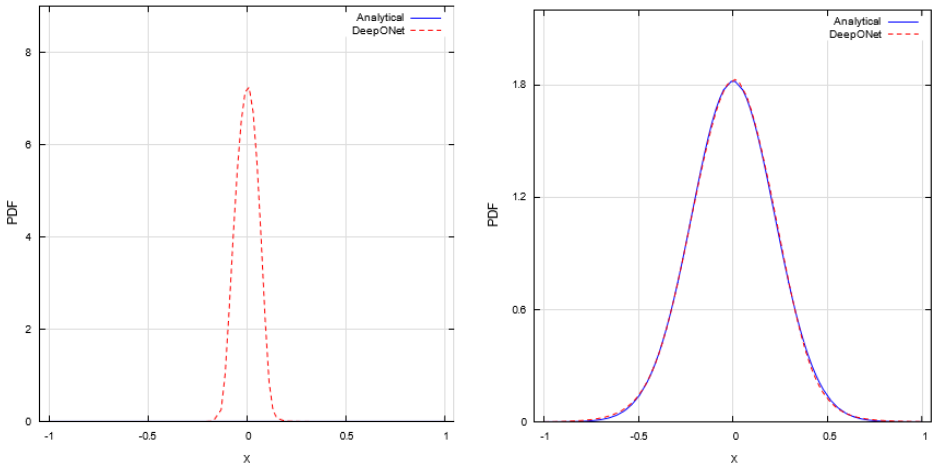


(c) $t = 0.25$ s (d) $t = 0.38$ s



(e) $t = 0.5$ s

Fig.6. Comparison between PINN and analytical probability density functions at different time steps



(a) $t = 0$ s (b) $t = 0.12$ s

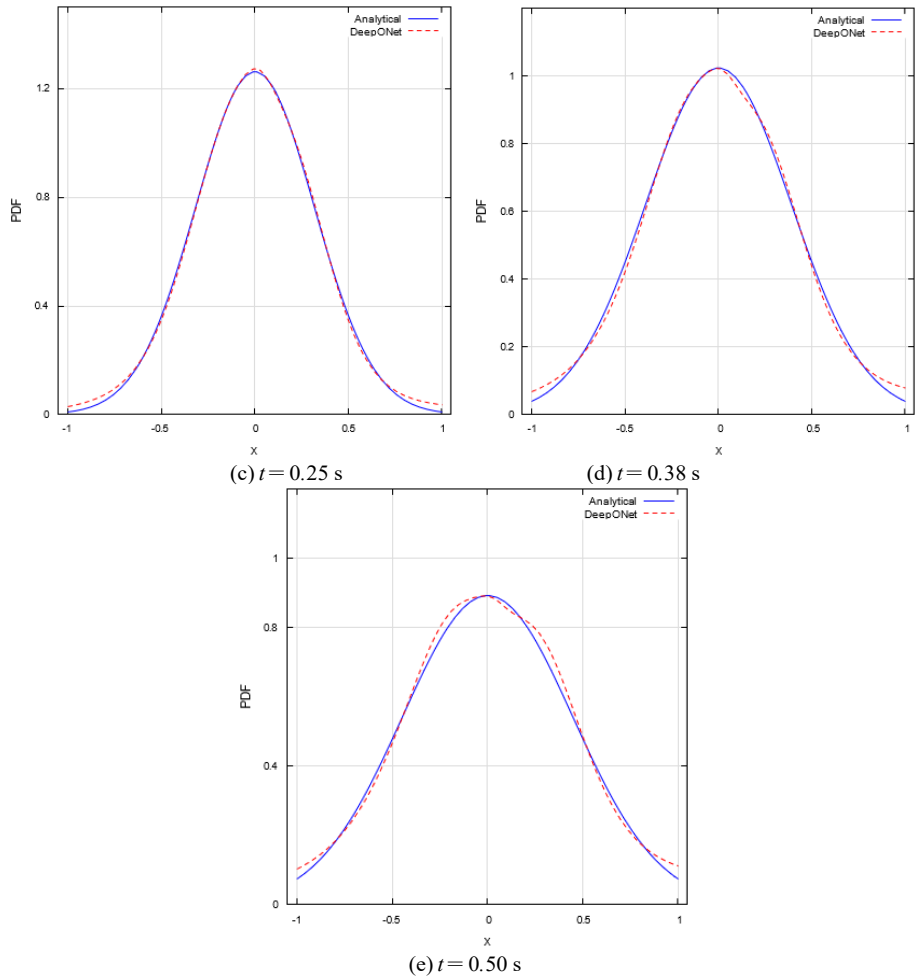


Fig.7. Comparison between DeepONet and analytical probability density functions at different time steps

Table 4. Probability conservation validation for PINN and DeepONet

Time (s)	PINN Integral	Analytical Integral	DeepONet Integral
0.00	1.0003	1.0000	1.0001
0.12	1.0002	1.0000	1.0008
0.25	1.0001	1.0000	1.0015
0.38	1.0001	1.0000	1.0021
0.50	1.0001	1.0000	1.0032

Table 5. Computational Cost Comparison of PINN and DeepONet

Metric	PINN	DeepONet
Epochs to Convergence	5000	1500
Training Time (hours)	2.15	0.85
Peak GPU Memory (GB)	4.8	3.2

While DeepONet converges faster (~1500 vs. 5000 epochs), it sacrifices physical fidelity, especially in sparse-data regions (e.g., $|x| > 0.7$ at $t = 0.5$ s). Quantitative metrics further support our approach:

- Error metrics: DeepONet's MSE (0.0582) and MAE (0.0232) far exceed PINN's (0.0018, 0.0085).
- Conservation: DeepONet shows up to 3.2% probability loss; PINN stays below 0.5%.
- Statistical validation: PINN maintains $R^2 = 0.994$ in MSD/VACF scaling, vs. DeepONet's $R^2 = 0.8177$.

While accuracy and physical consistency are paramount, the practical utility of a computational framework also depends on its efficiency. To provide a transparent analysis of the trade-offs involved, we measured the computational cost of training both the PINN and the DeepONet model. The results, summarized in Table 5, were obtained on a single NVIDIA A100 GPU.

Our physics-informed model required 5000 epochs to converge, with a total wall-clock training time of 2.15 hours and a peak GPU memory usage of 4.8 GB. In contrast, the purely data-driven DeepONet, which optimizes a much simpler loss function, converged in only 1500 epochs. This resulted in a significantly shorter training time of 0.85 hours and a lower memory footprint of 3.2 GB.

These results highlight a clear and important trade-off. The inclusion of physics-based loss terms, while crucial for achieving high accuracy and ensuring physical consistency, comes at a direct computational cost, making the PINN approximately 2.5 times slower to train than the DeepONet. This finding is critical for practitioners, as it suggests that for applications where speed is the absolute priority and some degree of physical inaccuracy is tolerable, a simpler data-driven model might be sufficient. However, for scientific simulations where physical laws must be respected and accuracy is non-negotiable, the additional computational expense of the PINN framework is justified by its vastly superior performance in producing physically reliable results.

Limitations include higher training cost, performance degradation beyond 10D (noted in [21]), and the need for careful calibration in recursive error bounds. Still, the trade-off favors physical accuracy over speed.

A mature scientific contribution requires a transparent acknowledgment of its boundaries and weaknesses. This section addresses several critical limitations of the current study.

- Model Assumptions: Our framework is built upon the Langevin-Fokker-Planck formalism. This assumes that the underlying stochastic process is Markovian (i.e., memoryless) and that the random forces are well-described by Gaussian white noise. The model would require significant modification, such as incorporating fractional derivatives or memory kernels into the PDE residual, to handle non-Markovian systems or those driven by non-Gaussian (e.g., Lévy) noise.
- Validation Scope and Scalability: The validation in this paper is performed on a low-dimensional (1D) benchmark problem. While this is a standard and necessary first step, the scalability of PINNs to high-dimensional systems remains a significant challenge due to the curse of dimensionality. As the number of dimensions increases, the number of collocation points required to adequately cover the domain grows exponentially, increasing computational cost. Future work must rigorously test this framework on higher-dimensional problems and compare its performance against specialized high-dimensional solvers, such as those based on tensor-train decompositions, which are known to be effective for certain classes of high-dimensional PDEs.
- Data Requirements: Our model was trained on high-quality, abundant synthetic data. This was a deliberate choice to isolate the performance of the learning algorithm itself. However, this means that the framework's performance in real-world, data-scarce scenarios, where experimental data may be sparse and noisy, remains an open and critical question. Investigating the model's sensitivity to data quality and quantity is a crucial next step for assessing its practical applicability.
- Practical Implementation Barriers: Several practical challenges were simplified in this study. First, PINNs can suffer from gradient pathologies, where imbalanced gradients between different loss terms can stall or destabilize training, particularly for stiff or multi-scale problems. Our use of manual loss weights and an adaptive learning rate is a heuristic solution; more advanced, adaptive weighting schemes could be necessary for more complex problems. Second, our use of a fixed-timestep Verlet integrator is a simplification. Implementing adaptive time-stepping with the Verlet algorithm is non-trivial and requires correction terms to maintain stability and accuracy, adding another layer of complexity for future development.

Compared to Chen et al. [16], who used networks to parametrize drift, our approach learns functional forms directly from Verlet-generated data and enforces physical consistency in extrapolation regimes. This enables applicability in noisy, real-world domains such as soft matter, financial systems, and plasma turbulence [1, 3].

Key extensions include:

- Modeling non-Markovian effects via memory kernels in the Fokker-Planck operator.
- Scaling to 100D benchmarks to test tractability in ultra-high-dimensional systems [21].
- Combining physics priors with deep generative models (e.g., score-based methods [19]) for efficiency gains.

The limitations discussed above naturally point toward a rich set of future research directions. Our framework provides a solid foundation upon which more sophisticated models can be built.

- Advanced Physical Metrics and Constraints: The framework could be enhanced by incorporating a wider range of physical constraints and validation metrics. For systems derived from a Hamiltonian, an energy conservation term could be added to the loss function to ensure the learned dynamics respect this fundamental principle, following recent work on Hamilton-Dirac Neural Networks. For non-equilibrium systems, one could calculate the entropy production rate from the learned PDF, providing a thermodynamic measure of the system's irreversibility and a deeper validation of its physical consistency.

- Efficient Sampling of Rare Events: A major challenge in stochastic simulations is the efficient sampling of rare but important events (e.g., protein folding, chemical reactions). Standard simulations can spend prohibitive amounts of time in metastable states. Future work could integrate our PINN framework with advanced rare-event sampling techniques, such as importance sampling or forward flux sampling. A PINN could be used to learn a biasing potential that accelerates the sampling of transition paths, dramatically improving computational efficiency for studying these critical phenomena.
- Hybrid Modeling and High-Dimensional Scaling: To tackle the scalability challenge, future research could explore hybrid approaches. For instance, one could combine our PINN framework with domain decomposition methods (like XPINNs) to enable parallel training on large domains. Another promising avenue is to hybridize PINNs with deep generative models, such as score-based models, which have shown success in high-dimensional density estimation, potentially leading to a more scalable and efficient framework for solving high-dimensional Fokker-Planck equations.

Overall, this work bridges data-driven adaptability with physics-grounded rigor, offering a scalable, robust tool for simulating complex stochastic systems.

7. Conclusions

This study introduces a robust Physics-Informed Neural Network (PINN) framework specifically designed for modeling Brownian motion, demonstrating enhanced accuracy and physical fidelity relative to classical numerical schemes and machine learning models such as DeepONet. By integrating Verlet-based trajectory generation with a multi-component loss function enforcing the Fokker-Planck equation, boundary conditions, and probability conservation, the proposed PINN achieves a mean squared error (MSE) of 0.0018 and mean absolute error (MAE) of 0.0085, markedly outperforming DeepONet's MSE of 0.0582 and MAE of 0.0232. The PINN maintains probability normalization within 0.03% across all time points, in contrast to DeepONet's deviations exceeding 0.32%, and accurately reproduces theoretical diffusive scaling (MSD = 2Dt) with an R^2 score of 0.994, confirming thermodynamic consistency. Recursive error bounding via secondary PINNs provides rigorous accuracy guarantees with worst-case errors below 1.2%, a capability absent in DeepONet. These results underscore the framework's ability to reconcile data-driven flexibility with physics-informed rigor, mitigating limitations of traditional solvers such as exponential complexity in high dimensions and numerical instabilities under sparse data, while addressing common shortcomings in existing PINNs, including unphysical artifacts and inadequate error quantification. Future work should extend the approach to non-Markovian systems exhibiting memory effects and improve scalability for ultra-high-dimensional stochastic processes through adaptive collocation or hybrid PINN-score-based architectures. Applications to real-world problems—such as financial volatility modeling and soft matter physics—alongside computational optimizations like reduced-order modeling and parallel training, could further enhance practicality and efficiency. By enforcing physical laws and quantifying errors systematically, this framework bridges stochastic dynamics and physics-informed learning, offering a scalable paradigm for robust scientific simulations.

Author Contributions Statement

Yulison Herry: Conceptualization, Methodology, and Supervision: Proposed the research idea, designed the overall physics-informed neural network framework for Brownian motion simulation, and supervised the project execution.
Julian Evan Chrisnanto: Data Curation and Software Implementation: Handled data generation via Verlet integration, dataset preprocessing, and implementing the PINN model in code.
Jeremia Oktavian Chrisnanto: Formal Analysis, Visualization, and Statistical Analysis: Performed in-depth analysis of experimental results, prepared performance charts, and ensured the statistical robustness of the evaluation.
Ferry Faizal: Writing – Review and Editing, and Project Management: Reviewed and edited the manuscript, ensured clarity and coherence, and coordinated project milestones and deadlines.

All authors have read and agreed to the published version of the manuscript.

Conflict of Interest Statement

The authors declare no conflict of interest.

Funding Declaration

This research received no external funding.

Data Availability Statement

The data and code used in this study are available upon reasonable request to the corresponding author. The simulation datasets generated via Verlet integration and the PINN training scripts can be shared for research reproducibility purposes.

Ethical Declarations
This study is purely computational and mathematical research involving no human subjects, animals, or sensitive personal data. Therefore, no ethical approval was required.

Acknowledgments
We would like to thank the referee for several valuable suggestions.

Declaration of Generative AI in Scholarly Writing
Authors should disclose in their article how AI and AI-assisted technologies were used during the writing process.

Abbreviations
The following abbreviations are used in this manuscript:
PINN - Physics-Informed Neural Network
SDE - Stochastic Differential Equation
PDE - Partial Differential Equation
FPE - Fokker-Planck Equation
ML - Machine Learning
DL - Deep Learning
NN - Neural Network
MSE - Mean Squared Error
MSD - Mean Squared Displacement
Adam - Adaptive Moment Estimation.

Appendix A: PINN Training Configuration Details
Table A1 provides the full hyperparameter configuration used during PINN training for the Brownian motion experiments reported in this study. All experiments were conducted using the Adam optimizer with the learning rate schedule and network architecture described in Section 3. The values below represent the final configuration after hyperparameter tuning.

Appendix B: Verlet Integration Algorithm Pseudocode
The velocity-Verlet integration scheme used for generating Brownian motion training trajectories follows the update rules described in Equations (5)–(8) of the main text. The pseudocode below summarises the numerical procedure implemented in Python for dataset generation:

Initialize: x_0, v_0, Δt, γ, k_B, T, m, N_steps
For each step n = 1 to N_steps:
    $\xi \leftarrow N(0, 1)$ [draw Gaussian noise]
    $F \leftarrow -\gamma \cdot v + \sqrt{(2\gamma k_B T/m)} \cdot \xi$ [Langevin force]
    $x_{n+1} \leftarrow x_n + v_n \cdot \Delta t + 0.5 \cdot (F/m) \cdot \Delta t^2$
    $v_{n+1} \leftarrow v_n + (F/m) \cdot \Delta t$
End For

References

[1] Niels Grønbech-Jensen and Oded Farago. A simple and effective Verlet-type algorithm for simulating Langevin dynamics. *Molecular Physics*, 111(8):983–991, April 2013.
[2] Tobias Gleim. Velocity-Verlet-like algorithm for simulations of stochastic dynamics, 2009. Version Number: 1.
[3] Chi Zhao, Feifei Zhang, Wenqiang Lou, Xi Wang, and Jianyong Yang. A comprehensive review of advances in physics-informed neural networks and their applications in complex fluid dynamics. *Physics of Fluids*, 36(10):101301, October 2024.
[4] Elodie Millan, Maxime Lavaud, Yacine Amarouchene, and Thomas Salez. Numerical simulations of con- fined Brownian-yet-non-Gaussian motion, 2022. Version Number: 2.
[5] NGUYEN Tue Tai. Numerical methods for simulating 1D Brownian motion and solving the Langevin equation. Course report (Statistical Machine Learning, f2023), Nagoya University, Jan. 2024. https://www.math.nagoya-u.ac.jp/~richard/teaching/f2023/SML_Tue_Tai_3.pdf
[6] Ryan D. Schumm and Paul C. Bressloff. A Numerical Method for Solving Snapping Out Brownian Motion in 2d Bounded Domains, 2023.
[7] Parul Tiwari, Don Kulasiri, and Sandhya Samarasinghe. Perspective Chapter: Numerical Solutions for Modelling Complex Systems with Stochastic Differential and Partial Differential Equations (SDEs/SPDEs). In Don Kulasiri, editor, *Stochastic Processes - Theoretical Advances and Applications in Complex Systems*. IntechOpen, June 2024.
[8] S. Bishnoi, Jayadeva, Sayan Ranu, and N. Krishnan. Graph Neural Stochastic Differential Equations for Learning Brownian Dynamics. *ArXiv*, abs/2306.11435, 2023.
[9] Lu Lu, Wenjie Yao, Francesc Verdugo, Steven G. Johnson, Raphae¨l Pestourie, and Zhicheng Wang. Physics- informed neural

networks with hard constraints for inverse design. *ArXiv*, abs/2102.04626, 2021.
[10] N. Evangelou, Felix Dietrich, J. M. Bello-Rivas, Alex J. Yeh, Rachel Stein, M. Bevan, and Ioannis G. Kevrekidis. Learning Effective SDEs from Brownian Dynamics Simulations of Colloidal Particles. *ArXiv*, abs/2205.00286, 2022.
[11] Xuhui Meng, Dongkun Zhang, Zhen Li, and George Em Karniadakis. PPINN: Parareal Physics-Informed Neural Network for time-dependent PDEs. *ArXiv*, abs/1909.10145, 2019.
[12] M. A. Nabian and H. Meidani. Adaptive Physics-Informed Neural Networks for Markov-Chain Monte Carlo. *ArXiv*, abs/2008.01604, 2020.
[13] Sai Ganga and Z. Uddin. Exploring Physics-Informed Neural Networks: From Fundamentals to Applications in Complex Systems. *ArXiv*, abs/2410.00422, 2024.
[14] Shambhu N. and Hiren G. The Fokker-Planck Equation. In Chris Myers, editor, *Stochastic Control*. Sciyo, August 2010.
[15] Chun-Wei Kong, Luca Laurenti, Jay W. McMahon, and Morteza Lahijanian. Error Bounds for Physics- Informed Neural Networks in Fokker-Planck PDEs. arXiv:2410.22371, 2024.
[16] Xiaoli Chen, Liu Yang, Jinqiao Duan, and G. Karniadakis. Solving Inverse Stochastic Problems from Discrete Particle Observations Using the Fokker-Planck Equation and Physics-informed Neural Networks. *ArXiv*, abs/2008.10653, 2020.
[17] Jared O'Leary, J. Paulson, and A. Mesbah. Stochastic physics-informed neural ordinary differential equa- tions. *J. Comput. Phys.*, 468:111466, 2022.
[18] Ruben Rodriguez-Torrado, Pablo Ruiz, L. Cueto-Felgueroso, M. Green, Tyler Friesen, S. Matringe, and Ju- lian Togelius. Physics-informed attention-based neural network for hyperbolic partial differential equations: application to the Buckley–Leverett problem. *Scientific Reports*, 12, 2022.
[19] Zhongqiang Zhang, Kenji Kawaguchi, Zheyuan Hu, and George Em Karniadakis. Score-Based Physics- Informed Neural Networks for High-Dimensional Fokker-Planck Equations. *ArXiv*, abs/2402.07465, 2024.
[20] David Kristjanson Duvenaud, Xuechen Li, Ting-Kam Leonard Wong, and Ricky T. Q. Chen. Scalable Gradients for Stochastic Differential Equations. *ArXiv*, abs/2001.01328, 2020.
[21] Taorui Wang, Zheyuan Hu, Kenji Kawaguchi, Zhongqiang Zhang, and G. Karniadakis. Tensor neural net- works for high-dimensional Fokker-Planck equations. *Neural networks: the official journal of the Interna- tional Neural Network Society*, 185:107165, 2025.
[22] M. Raissi, P. Perdikaris, and G. Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *J. Comput. Phys.*, 378:686–707, 2019.
[23] Qizhi He and Alexandre M. Tartakovsky. Physics-Informed Neural Network Method for Forward and Back- ward Advection-Dispersion Equations. *Water Resources Research*, 57, 2021.
[24] Rube´n Dar´ıo Ortiz Ortiz, Ana Magnolia Mar´ın Ram´ırez, and Miguel Á ngel Ortiz Mar´ın. Physics-Informed Neural Networks and Fourier Methods for the Generalized Korteweg–de Vries Equation. *Mathematics*, 13(9):1521, May 2025.
[25] M. Raissi, P. Perdikaris, and G. Karniadakis. Physics Informed Deep Learning (Part I): Data-driven Solutions of Nonlinear Partial Differential Equations. *ArXiv*, abs/1711.10561, 2017.
[26] G. Karniadakis, I. Kevrekidis, Lu Lu, P. Perdikaris, Sifan Wang, and Liu Yang. Physics-informed machine learning. *Nature Reviews Physics*, 3:422 – 440, 2021.
[27] S. Labed, N. Agram, and Bernt Oksendal. Fokker-Planck equations for McKean-Vlasov SDEs driven by fractional Brownian motion. arXiv:2409.07029, 2024.
[28] Zongren Zou and George Em Karniadakis. Multi-head physics-informed neural networks for learning functional priors and uncertainty quantification. Journal of Computational Physics, 531:113947, 2025.
[29] Amer Farea, Olli Yli-Harja, and Frank Emmert-Streib. Understanding Physics-Informed Neural Networks: Techniques, Applications, Trends, and Challenges. *AI*, 5(3):1534–1557, August 2024.
[30] Jose´ H. Blanchet and Xinyun Chen. Steady-state simulation of reflected Brownian motion and related stochastic networks. *Annals of Applied Probability*, 25:3209–3250, 2012.
[31] Jiachen Yao, Chang Su, Zhongkai Hao, Songming Liu, Hang Su, and Jun Zhu. MultiAdam: Parameter- wise Scale-invariant Optimizer for Multiscale Training of Physics-informed Neural Networks. *ArXiv*, abs/2306.02816, 2023.
[32] Roberto Molinaro, Siddhartha Mishra, and Tim De Ryck. wPINNs: Weak Physics informed neural networks for approximating entropy solutions of hyperbolic conservation laws. *ArXiv*, abs/2207.08483, 2022.

Authors' Profiles



Yulison Herry, received M.Sc. from Institute Technology Bandung (ITB), Indonesia. He is a As- sociate Professor at Jenderal of Achmad Yani University. His research interests include big data analysis, computational science, data mining. His work often bridges physics, informatics, and applied mathematics, as seen in collaborations with researchers from many universities and institutions in Indonesia.



Julian Evan received S.Sc. from Universitas Padjadjaran, Indonesia. He is currently a re- searcher at Universitas Padjadjaran since 2022. His research interests are computational mathematics, applying advanced techniques such as Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) models for a variety of complex problems.



Jeremia Oktavian is a researcher who has been affiliated with the Institute Technology Bandung (ITB) and is currently associated with the Graduate School of Life Science at Hokkaido University in Japan. His research appears to be in the field of life sciences, with a focus on microbiology, protein science, and molecular biology.



Ferry Faizal received M.Sc. from Kanazawa University, Japan, in the Division of Mathematics and Natural Sciences, concurrently with a Master's program in Computational Science at Institute Technology Bandung (ITB), Indonesia. Earned a Ph.D. from the Tokyo University of Agriculture and Technology (TUAT). His research interests are computational approaches to design, model, and predict the behavior of physical systems. Collaborative research has extended to other fields, such as fisheries, agriculture, and dentistry, particularly in applying physics-based methods to these domains.

How to cite this paper: Yulison Herry, Julian Evan, Jeremia Oktavian, Ferry Faizal, "PINNs for Stochastic Dynamics: Modeling Brownian Motion via Verlet Integration", International Journal of Information Technology and Computer Science(IJITCS), Vol.18, No.2, pp.126-145, 2026. DOI:10.5815/ijitcs.2026.02.08