PINNs for Non-Equilibrium Stochastic Systems

Ayush Ravi Chandran, Bach Luu, Deshna Singhal
Supervisor: Matthew Dobson

December 15, 2025

1 Introduction

The double-well potential is a standard benchmark for studying noisy dynamics, and therefore provides an ideal test problem for assessing how well physics-informed neural networks (PINNs) [3] and physics-informed normalizing flows (PINF) [2] solve Fokker-Planck equations. In this work, a one-dimensional double-well potential defines the underlying energy landscape for a stochastic particle system, and the associated Fokker-Planck equation for the probability density serves as the target partial differential equation. This setting combines bistability and noise-induced transitions between wells with a relatively simple structure that is convenient for numerical experiments.

In this work, we (i) benchmark Physics-Informed Normalizing Flows against classical stochastic simulation and finite-difference solvers on low-dimensional systems where ground truth is accessible, and (ii) identify key failure modes of PINF in multimodal, non-equilibrium settings such as the double-well potential, including instability in barrier regions and violation of probability mass conservation.

The double-well potential generates an ensemble of trajectories that concentrate near two stable states while occasionally crossing the barrier due to thermal fluctuations, leading to a stationary distribution with two pronounced peaks and a small but significant probability mass in the barrier region. Within this framework, the objective of the project is to compare how effectively PINNs and PINF approximate solutions of the Fokker-Planck equation associated with the chosen double-well system. The study examines the accuracy against classical numerical technique solutions.

We investigate the behavior of a particle in a one-dimensional double-well potential by two methods:

1. **Stochastic Differential Equation (SDE) Simulation:** We track the individual trajectories of a large number of particles governed by Stochastic Differential Equations. The simulation is advanced in time using the Euler-Maruyama method [1].
2. **Fokker-Planck Equation Solver:** We directly solve the corresponding Fokker-Planck equation, using the Crank-Nicolson finite difference scheme.

The position of the particle, $X(t)$, evolves according to the following stochastic differential equations:

- Double-well potential: $dX(t) = F(X(t))dt + \sigma dW_t$
  where:
  - $V(x) = (x^2 - 1)^2$ is the double-well potential.
  - $F(x) = -V'(x) = -4x(x^2 - 1)$
  - $\sigma = 1$ is the noise intensity, representing the strength of the thermal fluctuations.
  - dW_t is the increment of a Wiener process (Brownian motion), which is a Gaussian random variable with mean zero and variance dt. That is, $dW_t \sim \mathcal{N}(0, dt)$.
- Non-equilibrium systems: $dX(t) = -\nabla U(x)dt + \sigma dW_t + \eta dt$ on $[0, 4\pi]$
  where:

- $U(x) = \cos(x)$.
- $\sigma = 1$ is the noise intensity, representing the strength of the thermal fluctuations.
- $\eta = [-0.5, 0, 0.5]$ is a constant force.
- dW_t is the increment of a Wiener process (Brownian motion), which is a Gaussian random variable with mean zero and variance dt. That is, $dW_t \sim \mathcal{N}(0, dt)$.

The time evolution of the probability density function $p(x, t)$ for an ensemble of such particles is governed by the Fokker-Planck equation [5]:

$$\frac{\partial p(x,t)}{\partial t} = -\frac{\partial}{\partial x}[F(x)p(x,t)] + D\frac{\partial^2 p(x,t)}{\partial x^2}$$

where $D = \sigma^2/2$ is the diffusion coefficient. This equation describes the balance between the drift of probability due to the force $F(x)$ and the diffusion of probability due to random noise.

2 SDE Simulation with Euler-Maruyama Method

To simulate the stochastic dynamics of the system, we numerically solve the one-dimensional Itô stochastic differential equation (SDE) by discretizing time into small increments of size Δt. Since an exact analytical solution is generally unavailable, we employ the Euler-Maruyama method, which provides a first-order numerical approximation to the SDE and is widely used for simulating stochastic processes. Given a time step $t_{n+1} = t_n + \Delta t$ the Euler-Maruyama [1] update for the particle position is:

$$X_{n+1} = X_n + F(X_n)\Delta t + \sigma\sqrt{\Delta t}Z_n$$

where Z_n is a random number drawn from a standard normal distribution, $\mathcal{N}(0, 1)$. In our simulation, we track an ensemble of $N_p = 5000$ particles, starting from a uniform distribution in the range $[0, 2]$ for the double-well potential and $[0, \pi]$ for the non-equilibrium system. For the non-equilibrium system, we apply periodic wrapping so x stays in $[0, 4\pi]$. By evolving this ensemble forward in time using the Euler-Maruyama scheme, we obtain a numerical approximation to the time-dependent distribution of particle positions. These results can then be compared directly with solutions obtained from the Fokker-Planck equation, providing insight into both individual stochastic trajectories and the collective statistical behavior of the system.

3 Evolution Over Time

Instead of tracking individual sample paths of a one-dimensional Itô stochastic differential equation (SDE), we study the time evolution of the full probability distribution of the system. This distribution, denoted by p(x,t), describes the likelihood of finding the system at position x at time t. The evolution of this probability density is governed by the Fokker-Planck (FP) equation, which provides a deterministic description equivalent to the underlying stochastic dynamics. The Fokker-Planck equation captures two competing effects: advection of probability due to the deterministic drift and spreading of probability due to stochastic diffusion. Solving this equation allows us to observe how an initial distribution evolves over time toward equilibrium or other long-term behaviors. The probability density p(x, t) is governed by the Fokker-Planck equation associated with

$$\frac{\partial p(x,t)}{\partial t} = \mathcal{L}p(x,t)$$

where

$$\mathcal{L}p(x,t) = -\frac{\partial}{\partial x}(F(x)p(x,t)) + D\frac{\partial^2 p(x,t)}{\partial x^2} = \frac{\partial}{\partial x}(V(x)p(x,t)) + D\frac{\partial^2 p(x,t)}{\partial x^2}$$

We solve the Fokker-Planck equation by discretizing both space (grid points x_i) and time (steps t_n). First, we approximate the time derivative with a central difference around the half-step $t_{n+1/2}$

$$\frac{p(x_i, t_{n+1}) - p(x_i, t_n)}{\Delta t} = \frac{1}{2}[\mathcal{L}p(x_i, t_{n+1}) + \mathcal{L}p(x_i, t_n)]$$

After that, we spatially discretize $\mathcal{L}p(x, t)$ using a central difference for the space at a half-grid point $x_{i\pm1/2}$:

$$\begin{aligned}
\mathcal{L}p(x_i, t_n) =& \frac{[V(x_{i+1}) + V(x_i)][p(x_{i+1}, t_n) + p(x_i, t_n)] - [V(x_i) + V(x_{i-1})][p(x_i, t_n) + p(x_{i-1}, t_n)]}{2\Delta x} \\
& - D\frac{p(x_{i+1}, t_n) - 2p(x_i, t_n) + p(x_{i-1}, t_n)}{(\Delta x)^2} \\
=& \left(\frac{D}{(\Delta x)^2} - \frac{V(x_{i-1})}{2\Delta x}\right) p(x_{i-1}, t_n) - \left(\frac{2D}{(\Delta x)^2} + \frac{V(x_{i+1}) - V(x_{i-1})}{2\Delta x}\right) p(x_i, t_n) \\
& + \left(\frac{D}{(\Delta x)^2} + \frac{V(x_{i+1})}{2\Delta x}\right) p(x_{i+1}, t_n)
\end{aligned}$$

Then we can construct the tridiagonal matrix by collecting terms for p_{i-1}, p_i, and p_{i+1} gives a tridiagonal operator L.

Substituting the matrix L into the time-discretized scheme gives the final linear system to be solved at each time step:

$$\left(I - \frac{\Delta t}{2}L\right) p^{n+1} = \left(I + \frac{\Delta t}{2}L\right) p^n$$

3.1 Results and Discussion

The results from both simulation methods show excellent agreement with theoretical predictions.

The SDE simulation clearly illustrates the particle's behavior. The density heatmap (Figure 3) shows the initial distribution of particles (starting at $x \in [0, 2]$) immediately flowing into the nearest stable well at $x = 1$. Over time, the noise allows particles to cross the barrier, and the probability density spreads to populate both wells, eventually settling into the bimodal stationary distribution.

The Crank-Nicolson method (Figure 3) produces a similar heatmap. This plot is smoother because it represents a deterministic numerical approximation of the probability density evolution, free of sampling noise.

4 Steady-State Solution of SDE

The steady-state Solution of SDE refers to the probability density function such that:

$$\frac{\partial p(x, t)}{\partial t} = 0$$

indicating that the distribution does not change with time anymore and has reached equilibrium.

We know that for the SDE described in this paper the Steady-State solution is as follows:

$$p(x) \propto e^{-2/\sigma^2 V(x)}$$

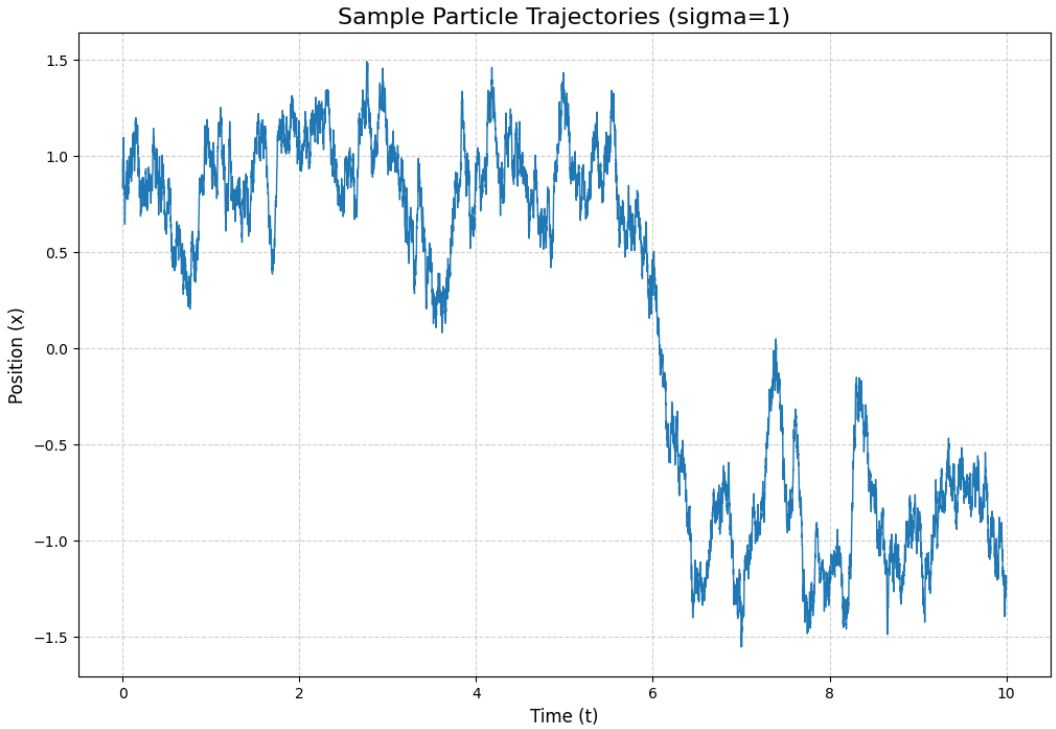

Figure 1: A sample trajectory from the SDE simulation for double-well. We can see it fluctuations near one well (around $x \approx \pm 1$) for long periods, with occasional noise-driven barrier crossing to the other well.

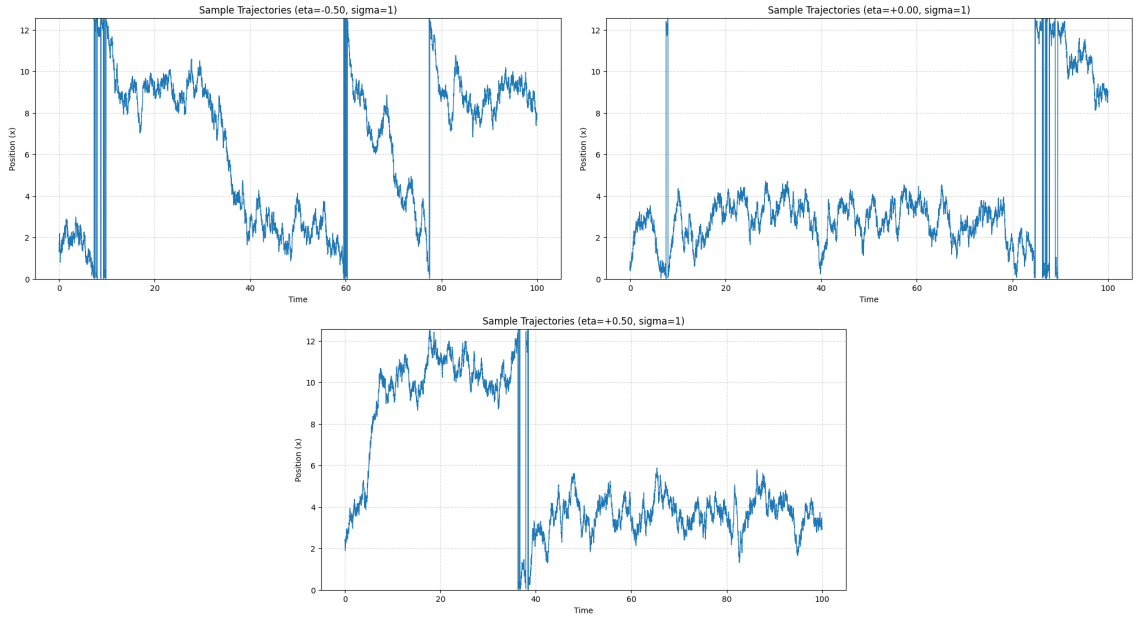

Figure 2: (Left) A sample trajectory from the SDE simulation for equation 2 ($\eta = -0.5$). (Right) A sample trajectory from the SDE simulation for equation 2 ($\eta = 0$). (Bottom) A sample trajectory from the SDE simulation for equation 2 ($\eta = 0.5$). Apparent discontinuous jumps correspond to periodic wrapping at the domain boundary.

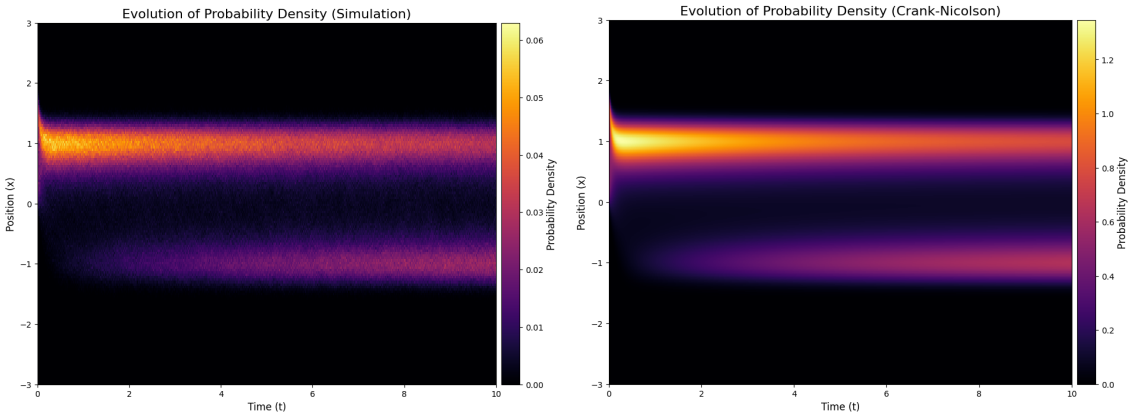

Figure 3: (Left) Heatmap of the empirical probability density evolution from the SDE simulation. (Right) Heatmap of probability density evolution from the Fokker-Planck equation integrated with the Crank-Nicolson method. The Crank-Nicolson [4] solution appears smoother because it evolves the density deterministically, without sampling noise.

Plugging the above equation into the FP equation we can verify this.

$$\begin{aligned}
&\frac{\partial}{\partial x}\big(V'(x)p(x)\big) + \tfrac{1}{2}\frac{\partial^2}{\partial x^2}\big(\sigma^2 p(x)\big) \\
&\quad= \frac{\partial}{\partial x}\Big(V'(x)p(x) + \tfrac{\sigma^2}{2}\frac{\partial p}{\partial x}\Big) \\
&\quad= \frac{\partial}{\partial x}\Big(V'(x)e^{-2V(x)/\sigma^2} + \tfrac{\sigma^2}{2}\frac{\partial}{\partial x}\big(e^{-2V(x)/\sigma^2}\big)\Big) \\
&\quad= \frac{\partial}{\partial x}\Big(V'(x)e^{-2V/\sigma^2} + \tfrac{\sigma^2}{2}\big(-\tfrac{2}{\sigma^2}V'(x)e^{-2V/\sigma^2}\big)\Big) \\
&\quad= \frac{\partial}{\partial x}\big(V'(x)e^{-2V/\sigma^2} - V'(x)e^{-2V/\sigma^2}\big) = \partial_x(0) = 0.
\end{aligned}$$

5 Physics-Informed Normalizing Flows for time-dependent Fokker-Planck equations with diffusion

We apply the Physics-Informed Normalizing Flows (PINF) method proposed by Liu and Wu [2] which can be used to transform the Fokker-Planck PDE into a system of characteristic ODEs solvable by neural networks. This mesh-free approach avoids spatial discretization entirely and preserves probability mass through the change-of-variables formula specified in the paper. It is also good at handling high dimensional problems, unlike traditional finite difference methods, which can be computationally expensive in higher dimensions.

The problem is set up as follows:

$$\begin{aligned}
\frac{\partial p(x,t)}{\partial t} &= -\nabla\cdot\big[p(x,t)\,\mu(x,t)\big] + \nabla\cdot\big[\nabla\cdot\big(p(x,t)\,D(x,t)\big)\big], \\
p(x,0) &= p_0(x).
\end{aligned}$$

where:

- $x \in \mathbb{R}^d$, $t \in [0,T]$
- $p(x,t) \geq 0$ is the probability density with $\int_{\mathbb{R}^d} p(x,t)dx = 1$ and $p(x,0) = p_0(x)$

- $\mu(x,t)$ is the drift.
- $D(x,t) \in \mathbb{R}^{d\times d}$ is the diffusion, with $D = \frac{1}{2}\sigma\sigma^T$

To validate our implementation of the Physics-Informed Normalizing Flows, we replicate the numerical experiment from Section 5.2 of [2]. This experiment demonstrates the PINF approach on a 10-dimensional TFP equation with both drift and diffusion terms.

5.1 Problem Formulation

We consider the TFP equation

$$\frac{\partial p}{\partial t} - \frac{1}{2}\Delta p + 2\nabla \cdot (p\mathbf{1}) = 0, \quad t \in [0,1],\, \boldsymbol{x} \in \mathbb{R}^{10} \tag{1}$$

with initial condition

$$\begin{aligned} p(\boldsymbol{x},0) &= (2\pi)^{-d/2}\exp(-\|\boldsymbol{x}\|^2/2) \\ &= (2\pi)^{-5}\exp(-\|\boldsymbol{x}\|^2/2) \end{aligned}$$

The exact solution is given by

$$p(\boldsymbol{x},t) = \frac{1}{(2\pi(t+1))^5}\exp\left(-\frac{\|\boldsymbol{x}-2t\cdot\mathbf{1}\|^2}{2(t+1)}\right) \tag{2}$$

To confirm this, we calculate the following quantities:

$$\begin{aligned} \nabla p = \frac{\partial p}{\partial x_i} &= p\cdot\frac{\partial p}{\partial x_i}\left(-\frac{\|\boldsymbol{x}-2t\cdot\mathbf{1}\|^2}{2(t+1)}\right) \\ &= p\cdot\left(-\frac{2(\boldsymbol{x}-2t\cdot\mathbf{1})}{2(t+1)}\right) \\ &= \frac{-p\cdot(\boldsymbol{x}-2t\cdot\mathbf{1})}{t+1} \end{aligned}$$

The Laplacian Δp can be calculated by:

$$\begin{aligned} \Delta p &= \sum_{i=1}^{10}\frac{\partial^2 p}{\partial x_i^2} \\ &= \sum_{i=1}^{10}\left[\frac{p(x_i-2t)^2}{(t+1)^2} - \frac{p}{(t+1)}\right] \\ &= \frac{p||\boldsymbol{x}-2t\cdot\mathbf{1}||^2}{(t+1)^2} - \frac{10p}{t+1} \end{aligned}$$

Finally,

$$\frac{\partial p}{\partial t} = p\left(\frac{-5}{t+1} + \frac{\|\boldsymbol{x}-2t\cdot\mathbf{1}\|^2}{2(t+1)^2} + \frac{2\cdot\mathbf{1}\cdot(\boldsymbol{x}-2t\cdot\mathbf{1})}{t+1}\right)$$

Substituting these values into our original TFP equation yields:

$$p\left(\frac{-5}{t+1} + \frac{\|\boldsymbol{x}-2t\cdot\mathbf{1}\|^2}{2(t+1)^2} + \frac{2\cdot\mathbf{1}\cdot(\boldsymbol{x}-2t\cdot\mathbf{1})}{t+1}\right) - \frac{1}{2}\left(\frac{p||\boldsymbol{x}-2t\cdot\mathbf{1}||^2}{(t+1)^2} - \frac{10p}{t+1}\right) - \frac{2p\cdot\mathbf{1}\cdot(\boldsymbol{x}-2t\cdot\mathbf{1})}{t+1} = 0$$

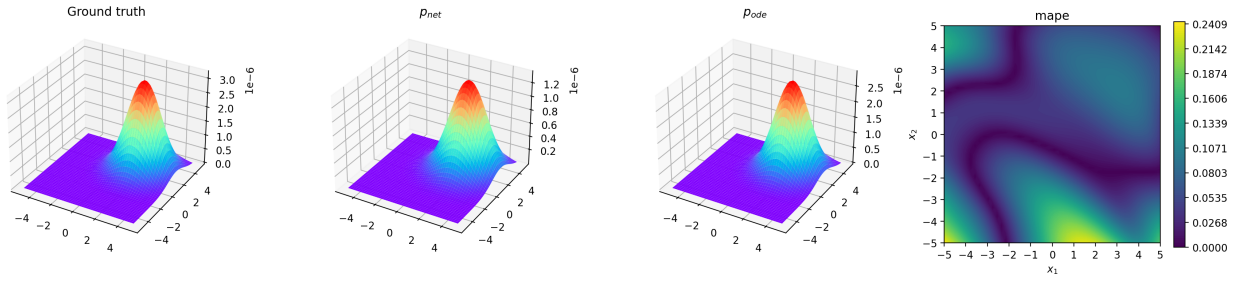

Figure 4: (Left) Exact and predicted solutions for the 10-dimensional TFP equation at $T = 1$. (Right) Corresponding MAPE distribution.

Algorithm 2 in the same paper allows us to transform our TFP equation into ODEs using the method of characteristics. We define an effective drift term μ^* given by

$$\mu^* = \mu - (\nabla \log p)\boldsymbol{D} - \nabla \cdot \boldsymbol{D}$$

where μ is the drift term and $\boldsymbol{D}$ is the diffusion term. For our equation, we have $\mu = 2 \cdot \mathbf{1}$ and $\boldsymbol{D} = \frac{1}{2}I$. So

$$\mu^* = 2 \cdot \mathbf{1} - \frac{1}{2}\nabla \log p$$

This yields the characteristic ODEs:

$$\frac{dx(t)}{dt} = 2 \cdot \mathbf{1} - \frac{1}{2}\nabla \log p$$
$$\frac{d \log p(x(t), t)}{dt} = -\nabla \cdot (\frac{d\boldsymbol{x}}{dt})$$

We implement the experiment using the code provided by the original authors[1]. The network architecture consists of $L = 4$ residual layers with $m = 32$ hidden neurons per layer, trained for $M = 10,000$ iterations using the Adam optimizer with learning rate $\eta = 0.01$ and batch size 2000. The characteristic ODEs are integrated using the Runge-Kutta (RK4) method with tolerance `atol`=10^{-8}, `rtol`=10^{-8}.

As described in the paper, for the activation function, we choose $\sigma(x) = |x| + \log(1 + e^{-2|x|})$, which has the derivative as $\sigma'(x) = \tanh(x)$. Unlike regular ResNets used for image recognition, we do not choose ReLU here. This is because PINFs require smooth, twice-differentiable activation functions. ReLU's derivative is piecewise constant with zero second derivative almost everywhere, making accurate divergence computation impossible.

5.1.1 Results

We evaluate the trained network at time $T = 1$ on a 50×50 uniform grid over the spatial domain $(\boldsymbol{x}_1, \boldsymbol{x}_2) \in [-5, 5]^2$. Figure 4 presents the analytical solution alongside the two PINF prediction outputs. The leftmost plot shows the ground truth solution stated above, which has Gaussian-like probability density centered near $\boldsymbol{x} = (2, 2, \ldots, 2)$. The drift term $2\mathbf{1}$ shifts the distribution from its initial center at the origin to $\mathbf{x} = (2, 2, \ldots, 2)$ over time $t = 1$, while diffusion causes the Gaussian to widen.

The middle two figures show the learned solutions from our neural network. The first evaluation $\hat{p}_{\text{net}}$ is computed by exponentiating the learned log-probability field $\phi_\theta(\mathbf{x}, t)$, while $\hat{p}_{\text{ode}}$ is obtained by solving the characteristic ODEs backward from time $t = 1$ to the initial condition at $t = 0$. We observe a good agreement between our ground truth and the network's predicted solution.

[1]Available at https://openreview.net/forum?id=uWHPW0sXFK

The rightmost figure showcases the spatial distribution of the Mean Absolute Percentage Error (MAPE) for the $\hat{p}_{\text{ode}}$ prediction. This is calculated as:

$$\text{MAPE}(\boldsymbol{x}) = \frac{|\hat{p}_{\text{ode}}(\boldsymbol{x}) - p_{\text{exact}}(\boldsymbol{x})|}{p_{\text{exact}}(\boldsymbol{x})} \times 100\% \tag{3}$$

where $\boldsymbol{x} = (\boldsymbol{x}_1, \boldsymbol{x}_2, 2, \ldots, 2) \in \mathbb{R}^{10}$.

The MAPE field shows maximum errors of approximately 24% concentrated along domain boundaries, with a diagonal pattern of moderate error across the domain. Lower errors in central regions indicate accurate prediction where probability mass is highest. This is consistent with the paper's result of the maximum error being 29% near the boundaries.

5.2 Double Well Potential

The double well system provides a challenging benchmark for PINF, testing its ability to capture bimodal distributions. Unlike the 10-dimensional TFP equation where classical methods become computationally prohibitive, the 1D double well allows direct comparison against traditional finite difference and Monte Carlo methods, enabling detailed analysis of PINF's strengths and limitations.

Applying the PINF algorithm to our FP equation with $\sigma = 0.5$, we define the effective drift as:

$$\mu^* = \mu - (\nabla \log p)D - \nabla \cdot D \tag{4}$$

For the double well system, $\mu(x) = 4x(1-x^2)$ and $D = \sigma^2/2 = 0.125$ is a constant scalar, giving $\nabla \cdot D = 0$. Therefore:

$$\mu^* = 4x(1 - x^2) - 0.125 \cdot \frac{\partial \log p}{\partial x} \tag{5}$$

This yields the characteristic ODEs:

$$\frac{dx(t)}{dt} = 4x(1 - x^2) - 0.125 \cdot \frac{\partial \log p}{\partial x} \tag{6}$$

$$\frac{d \log p(x(t), t)}{dt} = -\frac{\partial}{\partial x}\left(\frac{dx}{dt}\right) \tag{7}$$

where $\log p(x, t)$ is parameterized by the neural network $\phi_\theta(x, t)$. The initial condition here is a Mixed Gaussian consisting of an equal number of particles centered at -1 and 1. We use the same activation function as described in section 5.1. The network consists of $L = 3$ residual layers with $m = 64$ hidden neurons per layer, trained for $M = 10,000$ epochs using the Adam optimizer and a learning rate of 5×10^{-4} and batch size 512. The characteristic ODEs are integrated using the Dormand-Prince adaptive Runge-Kutta method of order 5 (dopri5) with tolerances `atol`=10^{-3}, `rtol`=10^{-3}.

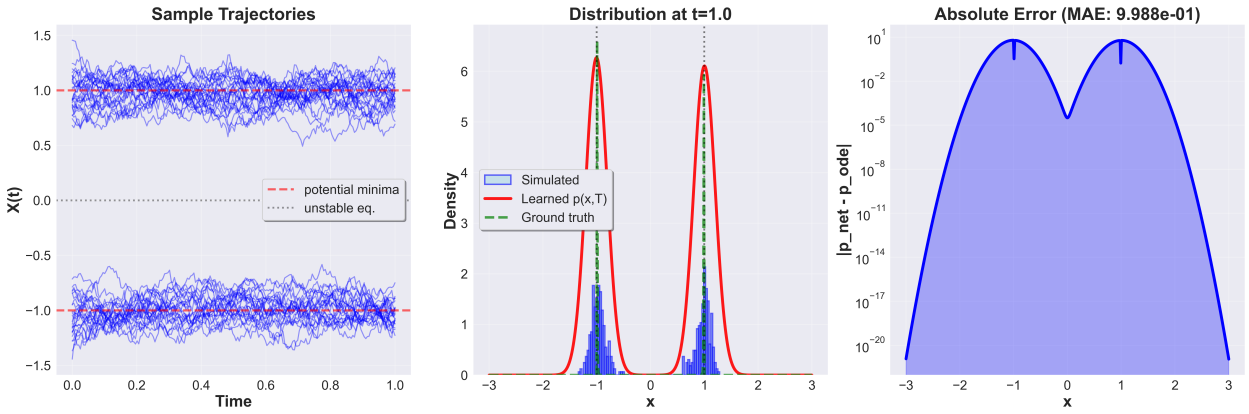

Figure 5: PINF results at $T = 1$. (Left) Sample trajectories showing particles trapped near $x = \pm 1$ without well-to-well transitions. (Middle) Learned density $\hat{p}_{\text{net}}$ (red) and ground truth $\hat{p}_{\text{ode}}$ (green) both overestimate the empirical distribution (blue histogram). (Right) Absolute error on log scale, MAE ≈ 1.0.

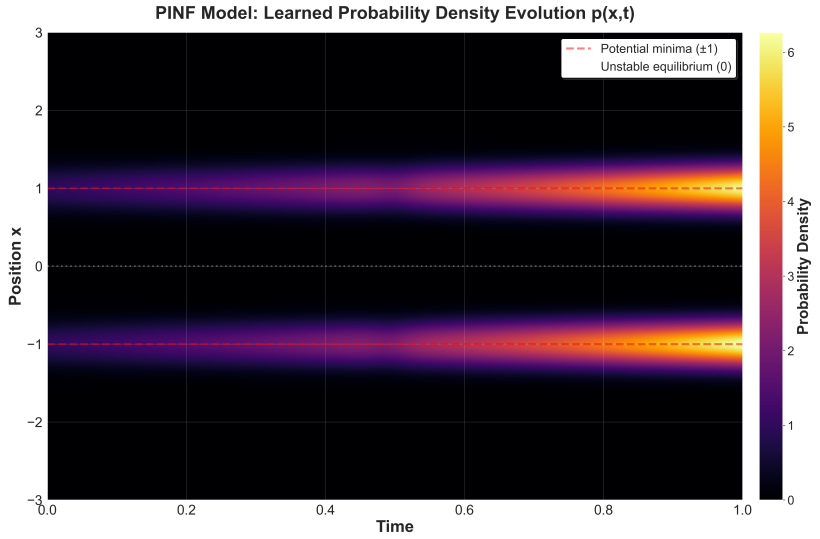

Figure 6: Heatmap of the evolution of the probability density computed from the PINF method.

Figure 6 displays the space-time evolution learned by PINF. Two bands of high probability persist at $x = \pm 1$ throughout $t \in [0, 1]$, matching the expected behavior of particles trapped near potential minima due to the specific initial condition used. However, the space-time evolution exhibits several problems. The probability bands expand over time for both the minima, which suggests that the learned density violates the fundamental constraint $\int p(x, t)\, dx = 1$, indicating that the PINF formulation does not inherently enforce mass conservation in this setting. Since the total probability must integrate to 1 at all times, we see that the network fails to maintain the normalization constraint. While SDE simulations provide particle-level intuition and Crank-Nicolson yields stable and accurate density evolution, PINF offers scalability at the cost of stability and accuracy in multimodal regimes.

The heatmap also shows no evidence of particles transitioning between the two wells. In the true dynamics, thermal fluctuations cause occasional barrier crossings between $x = -1$ and $x = 1$, which should appear as faint connecting regions of non-zero probability. This behavior is consistent with the structure of the learned score function $\nabla \log p$, which becomes unreliable in low-density regions such as the barrier be-

tween wells. As a result, the characteristic ODEs fail to transport probability mass across modes, effectively decoupling the dynamics of each well.

The observed limitations suggest several concrete directions for improvement. In particular, more accurate evaluation of the characteristic ODEs, tighter integration tolerances, and longer training horizons may improve stability and enable the model to better resolve fine-scale features such as barrier-crossing dynamics. Additionally, incorporating explicit normalization or mass-conservation constraints could address the observed drift in total probability.

6 Conclusion

In this work, we compared stochastic simulation, classical numerical PDE solvers, and neural approaches for solving Fokker-Planck equations arising from non-equilibrium systems. The Euler-Maruyama and Crank-Nicolson methods produced consistent and accurate results, with the latter providing smooth, deterministic evolution of the probability density. Physics-Informed Normalizing Flows (PINF) successfully captured the dynamics in high-dimensional Gaussian settings, demonstrating their scalability and mesh-free advantages.

However, in the double-well system, PINF struggled to represent multimodal behavior, failing to capture barrier-crossing dynamics and maintain probability normalization. This highlights a key limitation of neural PDE solvers in settings with complex energy landscapes and sharp transitions. Our results suggest that while PINF is promising for high-dimensional problems, classical methods remain more reliable for low-dimensional systems with intricate structure. Future work should focus on improving stability, enforcing conservation constraints, and better handling multimodality in neural approaches.

References

[1] D. J. Higham, *An algorithmic introduction to numerical simulation of stochastic differential equations*, SIAM Review, 43 (2001), pp. 525–546.

[2] F. Liu, F. Wu, and X. Zhang, *PINF: Continuous Normalizing Flows for Physics-Constrained Deep Learning*, arXiv:2309.15139, (2023).

[3] M. Raissi, P. Perdikaris, and G. E. Karniadakis, *Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations*, Journal of Computational Physics, 378 (2019), pp. 686–707.

[4] Wikipedia contributors, *Crank Nicolson Method, Wikipedia, The Free Encyclopedia*, 2025. [Online; accessed 18-December-2025].

[5] ——, *Fokker–Planck equation, Wikipedia, The Free Encyclopedia*, 2025. [Online; accessed 18-December-2025].