Journal of Machine Learning **ISSN**: 2790-2048(e), 2790-203X(p)

Solving Bivariate Kinetic Equations for Polymer Diffusion Using Deep Learning

Heng Wang * [1] and Weihua Deng † [1]

[1]School of Mathematics and Statistics, Gansu Key Laboratory of Applied Mathematics and Complex Systems, Lanzhou University, Lanzhou 730000, China.

Abstract. In this paper, we derive a class of backward stochastic differential equations (BSDEs) for infinite-dimensionally coupled nonlinear parabolic partial differential equations, thereby extending the deep BSDE method. In addition, we introduce a class of polymer dynamics models that accompany polymerization and depolymerization reactions, and derive the corresponding Fokker-Planck equations and Feynman-Kac equations. Due to chemical reactions, the system exhibits a Brownian yet non-Gaussian phenomenon, and the resulting equations are infinitely dimensionally coupled. We solve these equations numerically through our new deep BSDE method, and also solve a class of high-dimensional nonlinear equations, which verifies the effectiveness and shows approximation accuracy of the algorithm.

Keywords:
BSDEs,
Deep BSDE method,
Polymer dynamics,
Brownian yet non-Gaussian.

Article Info.:
Volume: 3
Number: 2
Pages: 215 - 244
Date: /2024
doi.org/10.4208/jml.240124

Article History:
Received: 24/01/2024
Accepted: 09/04/2024

Communicated by:
Jiequn Han

1 Introduction

In recent years, the diffusion phenomenon with non-Gaussian shape in complex systems has been gradually discovered experimentally, e.g. microbeads in lipid tubes [36], networks [37] or in a matrix of micropillars [4]. More examples include the movement of tracers in colloids, polymeric, and active suspensions [39], and the motion of individuals in heterogeneous populations such as nematodes [20]. This paper develops the deep BSDE method to model and simulate this kind of polymer dynamics.

More concretely, we focus on the diffusion behavior of a class of polymeric microparticles that accompany polymerization and depolymerization chemistry. It is shown that the polymerization and depolymerization of molecules is the natural basis of Brownian yet non-Gaussian diffusion of the center of mass (CM) [1]. The diffusivity D of the CM is affected by the molecular size N, and the molecular size is determined by the chemical reaction and changes randomly, so the diffusivity $D(N(t))$ becomes a stochastic process. Such random diffusivity leads to Brownian yet non-Gaussian diffusion. Further, reference [27] discusses the motion of a polymer in a chemostatted monomer bath while the monomer concentration in the bath changes. Reference [41] derives the Fokker-Planck equations for

*2202209341610lzu.edu.cn
†Corresponding author. dengwh@lzu.edu.cn

CM and the Feynman-Kac equations through subordination technique [3, 5, 13]. Since the size of the polymer particle is a discrete variable, the resulting equation is of bivariate form

$$\frac{\partial}{\partial t}u(n,x,t) = \mathcal{T}_n u(n,x,t) + \frac{1}{2}\mathrm{Tr}\big(\sigma\sigma^\top(n,x,t)(\mathrm{Hess}_x)u(n,x,t)\big) + \nabla_x u(n,x,t)\cdot\mu(n,x,t) + f\big(t,n,x,u(n,x,t),(\sigma^\top\nabla_x u)(n,x,t)\big) \tag{1.1}$$

with the initial condition $u(n,x,0) = g(n,x)$, here $\mathcal{T}_n$ is an operator with respect to the discrete variable n. If we consider n as a state parameter, then Eq. (1.1) can be seen as an infinite-dimensionally coupled system, the way of coupling depends on the operator $\mathcal{T}_n$.

With the growth of data resources and computing power, deep learning has been becoming an important methodology of our research. In recent years, a large number of deep learning-based partial differential equation (PDE) solvers have been developed, most of which are inspired by traditional methods. But unlike traditional methods, learning-based methods reduce the requirements for meshing and directly use neural networks as basis functions. These improvements allow us to avoid various complex problems encountered by traditional methods when solving PDEs. For example, the deep Ritz method [12] is a deep learning method based on the variational principle, which uses deep learning to solve the variational problem corresponding to PDEs. Least squares-based deep learning methods include deep Galerkin method [34] and physics-informed neural networks [32], which train models by minimizing the squared residuals of PDEs. Physics-informed neural networks also has a discrete-time version, which is based on the Runge-Kutta method. Weak adversarial networks [40] provide a method for solving the weak formulations of high-dimensional partial differential equations through adversarial learning. E *et al.* [11, 18] propose a deep learning method for solving parabolic PDEs based on BSDEs, called the deep BSDE method. References [15, 19] provide posterior estimates of the deep BSDE method.

The main contributions of this paper are as follows. Defining

$$\mathcal{T}_n f(n) = \begin{cases} \alpha(n)\big(f(n+1)-f(n)\big) + \beta(n)\big(f(n-1)-f(n)\big), & n \geq 1, \\ \alpha(0)\big(f(1)-f(0)\big), & n = 0, \end{cases} \tag{1.2}$$

where $\alpha(n)$ and $\beta(n)$ are known functions, we derive the BSDEs of Eq. (1.1) by constructing a stochastic process $X(t)$ coupled with the birth-death process $N(t)$ and extend the deep BSDE method (see Section 2). In addition, we present a class of applications of our new method in solving polymer dynamics problems. Specifically, we model a class of polymer particle diffusion dynamics accompanied by polymerization and depolymerization reactions, and derive the forward (backward) Fokker-Planck equations and the corresponding Feynman-Kac equations (see Section 3). To solve the Feynman-Kac equations, we extend the deep BSDE method to the space of frequency domain (see Section 4). On the aspect of the deep BSDE method, the main distinction of this work from [18] can be summarized as follows:

(i) The process considered in this paper is the coupling of the diffusion process with the jumping one, instead of the pure diffusion process.

(ii) The developed method is to solve a system coupling high-dimensional system with infinite-dimensional lattice system, and an appropriate neural network structure is carefully designed (see Section 5).

(iii) The BSDE derived in this paper is different from the one in references [29, 30], since it has a jumping term. Then, most of existing conclusions are no longer applicable.

In the following, we first derive the BSDEs of Eq. (1.1) and design the corresponding deep BSDE method. In Section 3, starting from the Langevin equation, we introduce a class of polymer dynamics models that accompany chemical reactions and derive the corresponding Fokker-Planck equations and Feynman-Kac equations. In Section 4, we perform extensively numerical experiments for the specific application problems obtained in Section 3, and designed and solved a class of high-dimensional and nonlinear problems, which verifies the effectiveness of our new algorithm in solving various problems. In Section 5, we give the details of the neural network structures, and verify the performance of the neural networks and the factors affecting the accuracy of the algorithm through numerical experiments. At the end, we summarize our work and discuss the remaining issues of this method to be solved and our future work.

2 BSDEs and deep BSDE method

The deep BSDE method [11, 18] is an effective deep learning method for solving high-dimensional semi-linear parabolic PDEs. The key idea of this method is to use the connection between PDEs and BSDEs [29, 30] to transform the problem of solving PDEs into the one of solving the corresponding BSDEs. In this section, we derive the BSDEs satisfied by the solution of Eq. (1.1) and design the numerical scheme of our new deep BSDE method.

2.1 BSDEs

Without lose of generality, we consider a class of semilinear parabolic PDEs with infinite dimensional coupling. These PDEs can be represented as

$$\begin{aligned}&\frac{\partial}{\partial t}u(n,x,t)+\mathcal{T}_n u(n,x,t)+\mathcal{T}_x u(n,x,t)\\&\quad+f\big(t,n,x,u(n,x,t),\big(\sigma^\top\nabla_x u\big)(n,x,t)\big)=0\end{aligned}\tag{2.1}$$

with the given terminal condition $u(n,x,T)=g(n,x)$. Here the unknown is $u:\mathbb{N}\times\mathbb{R}^d\times[0,\infty)\to\mathbb{R}$, $\mathcal{T}_n$ is a difference operator on variable n defined by (1.2),

$$\mathcal{T}_x=\frac{1}{2}\sum_{i=1}^{d}(\sigma\sigma^\top)_{ii}(n,x,t)\frac{\partial^2}{\partial x_i^2}+\sum_{i=1}^{d}\mu_i(n,x,t)\frac{\partial}{\partial x_i},\tag{2.2}$$

μ is a known vector-valued function, σ is a known $d\times d$ matrix-valued function, and f is a known nonlinear function.

Let $N(t)$ be a birth-death process that satisfies

$$\begin{aligned}&\mathbb{P}\big(N(t+\tau)-N(t)=k|N(t)=n\big)\\&=\begin{cases}\alpha(n)\tau+o(\tau), & k=1,\\ \beta(n)\tau+o(\tau), & k=-1,\\ 1-\big(\alpha(n)+\beta(n)\big)\tau+o(\tau), & k=0,\\ o(\tau), & \text{otherwise,}\end{cases}\end{aligned}\tag{2.3}$$

where $\alpha(n),\beta(n)\geq 0$ for $n\in\mathbb{N}$, and $\beta(0)=0$. Let $\Delta N(t;N(t-))=N(t)-N(t-)$ be the jump process with respect to $N(t)$. For any subset $A\subseteq\mathbb{Z}\setminus\{0\}$, define the counting random measure

$$J\big(dt,A;N(t-)\big)=\sum_{n\in A}\delta\big(\Delta N\big(t;N(t-)\big)-n\big)dt.\tag{2.4}$$

Then the birth-death process $N(t)$ can be re represented as an integral form

$$N(t)=N(0)+\int_0^t\int_{\mathbb{Z}\setminus\{0\}}nJ\big(d\tau,dn;N(\tau-)\big).\tag{2.5}$$

Let $\nu(dt,A;N(t-))=\pi(A;N(t-))dt$ be a compensator of J such that

$$\widetilde{J}\big(t,A;N(t-)\big)=\int_0^t\int_A(J-\nu)\big(d\tau,dn;N(\tau-)\big)$$

is a martingale. Note here,

$$\pi(\{k\};n)=\begin{cases}\alpha(n), & k=1,\\ \beta(n), & k=-1,\\ 0, & \text{otherwise.}\end{cases}\tag{2.6}$$

Let $B(t)$ be a d-dimensional Brownian motion and $X(t)$ be a d-dimensional stochastic process satisfying

$$X(t)=X(0)+\int_0^t\mu\big(N(\tau),X(\tau),\tau\big)d\tau+\int_0^t\sigma\big(N(\tau),X(\tau),\tau\big)dB(\tau).\tag{2.7}$$

Driven by the stochastic processes $N(t)$ and $X(t)$, the following result holds.

Lemma 2.1. *If $X(t)$ is a stochastic integral of the form* (2.7)*, then, for each $u(n,\cdot,\cdot)\in C^2(\mathbb{R}^d)\times C^1([0,\infty))$, $t\geq 0$, we have*

$$\begin{aligned}&u\big(N(t),X(t),t\big)-u\big(N(0),X(0),0\big)\\&=\int_0^t\frac{\partial}{\partial t}u\big(N(\tau),X(\tau),\tau\big)+\mathcal{T}_xu\big(N(\tau),X(\tau),\tau\big)d\tau+\int_0^t\big(\nabla_xu^\top\sigma\big)\big(N(\tau),X(\tau),\tau\big)dB(\tau)\\&\quad+\int_0^t\int_{\mathbb{Z}\setminus\{0\}}u\big(N(\tau-)+n,X(\tau),\tau\big)-u\big(N(\tau-),X(\tau),\tau\big)J\big(d\tau,dn;N(\tau-)\big).\end{aligned}\tag{2.8}$$

Proof. Define a sequence of stopping times recursively by $T_0 = 0$ and

$$T_n = \inf\{t > T_{n-1}; |N(t) - N(T_{n-1})| \neq 0\}.$$

For each $t > 0$, we have

$$\begin{aligned}
&u\big(N(t),X(t),t\big) - u\big(N(0),X(0),0\big)\\
&= \sum_{j=0}^{\infty} u\big(N(t\wedge T_{j+1}),X(t\wedge T_{j+1}),t\wedge T_{j+1}\big) - u\big(N(t\wedge T_j),X(t\wedge T_j),t\wedge T_j\big)\\
&= \sum_{j=0}^{\infty} u\big(N(t\wedge T_{j+1}-),X(t\wedge T_{j+1}),t\wedge T_{j+1}\big) - u\big(N(t\wedge T_j),X(t\wedge T_j),t\wedge T_j\big)\\
&\quad + \sum_{j=0}^{\infty} u\big(N(t\wedge T_{j+1}),X(t\wedge T_{j+1}),t\wedge T_{j+1}\big) - u\big(N(t\wedge T_{j+1}-),X(t\wedge T_{j+1}),t\wedge T_{j+1}\big)\\
&= \int_0^t \Big\{ \frac{\partial}{\partial t} u\big(N(\tau),X(\tau),\tau\big) + \frac{1}{2}\mathrm{Tr}\big[(\sigma\sigma^\top)\big(N(\tau),X(\tau),\tau\big)(\mathrm{Hess}_x)u\big(N(\tau),X(\tau),\tau\big)\big]\\
&\qquad + (\nabla_x u\cdot\mu)\big(N(\tau),X(\tau),\tau\big)\Big\} d\tau\\
&\quad + \int_0^t \big[\nabla_x u\big(N(\tau),X(\tau),\tau\big)\big]^\top \sigma\big(N(\tau),X(\tau),\tau\big) dB(\tau)\\
&\quad + \int_0^t \int_{\mathbb{Z}\setminus\{0\}} u\big(N(\tau-)+n,X(\tau),\tau\big) - u\big(N(\tau-),X(\tau),\tau\big) J\big(d\tau,dn;N(\tau-)\big),
\end{aligned}$$

where we have used the Itô's formula. □

With the help of Lemma 2.1, we can easily obtain the BSDE satisfied by the solution $u(n,x,t)$ of Eq. (2.1).

Theorem 2.1. *Let $u(n,x,t)$ be the solution of the Eq.* (2.1)*, $N(t)$ be a stochastic process satisfying* (2.3)*, and $X(t)$ be a stochastic process satisfying* (2.7)*. Then $u(N(t),X(t),t)$ satisfies the BSDE*

$$\begin{aligned}
&u\big(N(t),X(t),t\big) - g\big(N(T),X(T)\big)\\
&= \int_t^T f\Big(\tau,N(\tau),X(\tau),u\big(N(\tau),X(\tau),\tau\big),\big(\sigma^\top\nabla_x u\big)\big(N(\tau),X(\tau),\tau\big)\Big) d\tau\\
&\quad - \int_t^T \int_{\mathbb{Z}\setminus\{0\}} u\big(N(\tau-)+n,X(\tau),\tau\big) - u\big(N(\tau-),X(\tau),\tau\big)\widetilde{J}\big(d\tau,dn;N(\tau-)\big)\\
&\quad - \int_t^T \big[\nabla_x u\big(N(\tau),X(\tau),\tau\big)\big]^\top \sigma\big(N(\tau),X(\tau),\tau\big) dB(\tau). \qquad (2.9)
\end{aligned}$$

Proof. Applying Lemma 2.1 to $u(N(s),X(s),s)$ between $s = t$ and $s = T$, one can get the BSDE (2.9). □

Let $N_s^{n,t} = N(s)$ be a birth-death process with $N(t) = n$ and $X_s^{x,t} = X(t)$ the corresponding subordinated stochastic process with $X(t) = x$. One can get a direct conclusion through Theorem 2.1.

Corollary 2.1. *If $f(t,n,x,y,z) = c(t,n,x)y$, then the BSDE* (2.9) *has the explicit solution*

$$\begin{aligned} & u\big(N_s^{n,t}, X_s^{x,t}, s\big) - e^{\int_s^T c(\tau, N_\tau^{n,t}, X_\tau^{x,t})d\tau} g\big(N_T^{n,t}, X_T^{x,t}\big) \qquad (2.10)\\ = & -\int_s^T e^{\int_s^\tau c(r, N_r^{n,t}, X_r^{x,t})dr} \big(\nabla_x u^\top \sigma\big)\big(N_\tau^{n,t}, X_\tau^{x,t}, \tau\big) dB(\tau) \\ & -\int_s^T e^{\int_s^\tau c(r, N_r^{n,t}, X_r^{x,t})dr} \int_{\mathbb{Z}\setminus\{0\}} u\big(N_{\tau-}^{n,t} + n, X_\tau^{x,t}, \tau\big) - u\big(N_{\tau-}^{n,t}, X_\tau^{x,t}, \tau\big)\widetilde{J}\big(d\tau, dn; N_{\tau-}^{n,t}\big), \end{aligned}$$

and

$$u(n,x,t) = \mathbb{E}\big[u\big(N_t^{n,t}, X_t^{x,t}, t\big)\big] = \mathbb{E}\Big[e^{\int_t^T c(\tau, N_\tau^{n,t}, X_\tau^{x,t})d\tau} g\big(N_T^{n,t}, X_T^{x,t}\big)\Big]. \qquad (2.11)$$

2.2 Deep BSDE method

We focus on the solution $u(n_0, x_0, 0)$, where $(n_0, x_0) \in \mathbb{N} \times \mathbb{R}^d$ is already determined. We treat $u(n_0, x_0, 0) \approx \theta_\phi$[1] as a parameter in the model and view BSDE (2.9) as the way to get the value of u at the terminal time T when $u(N(0) = n_0, X(0) = x_0, 0)$ and $\nabla_x u(N(t), X(t), t)$ and $u(N(t) \pm 1, X(t), t)$ are known, where $(\sigma^\top \nabla_x u)(n,x,t)$ is approximated by a neural network

$$\big(\sigma^\top \nabla_x u\big)(n,x,t) \approx \psi_1(n,x,t|\theta_{\psi_1}) \qquad (2.12)$$

with parameters θ_{ψ_1}, and

$$\delta_n^\pm u(n,x,t) = [u(n+1,x,t) - u(n,x,t), u(n-1,x,t) - u(n,x,t)]^\top$$

is approximated by a neural network

$$\delta_n^\pm u(n,x,t) \approx \psi_2(n,x,t|\theta_{\psi_2}) \qquad (2.13)$$

with parameters θ_{ψ_2}[2].

On this basis, one can design the numerical schemes by discretizing time. Given a partition of the time interval $[0,t] : 0 = t_0 < t_1 < \cdots < t_{N-1} < t_N = T$, we consider the simple Euler scheme for $k = 0, 1, \ldots, N-1$,

$$X(t_{k+1}) - X(t_k) = \mu\big(N(t_k), X(t_k), t_k\big)\Delta t_k + \sigma\big(N(t_k), X(t_k), t_k\big)\Delta B(t_k), \qquad (2.14)$$

[1]Here, we design our algorithm by taking the solution of Eq. (2.1) at a fixed time $t = 0$ and a single space-point (n_0, x_0) as an example, so that the target solution can be approximated by a parameter denoted as θ_ϕ. When one is interested in the solution of a region, one can also use the proposed method, as described in Section 5.1, where one only needs to use a neural network to approximate the target solution $u(n,x,0) \approx \phi(n,x;\theta_\phi)$.

[2]Here, in order to make the formulation of the algorithm clearer, we use two different neural networks, $\psi_1(n,x,t)$ and $\psi_2(n,x,t)$, to approximate $(\sigma^\top \nabla_x u)(n,x,t)$ and $\delta_n^\pm u(n,x,t)$, respectively. Obviously, one can also merge them into a single neural network $\psi : \mathbb{N} \times \mathbb{R}^d \times [0,\infty) \to \mathbb{R}^{d+2}$.

$$
\begin{aligned}
&u\big(N(t_{k+1}),X(t_{k+1}),t_{k+1}\big)\\
&=u\big(N(t_k),X(t_k),t_k\big)+\big[\nabla_x u\big(N(t_k),X(t_k),t_k\big)\big]^\top\sigma\big(N(t_k),X(t_k),t_k\big)\Delta B(t_k)\\
&\quad+u\big(N(t_{k+1}),X(t_k),t_k\big)-u\big(N(t_k),X(t_k),t_k\big)-\mathcal{T}_{N(t_k)}u\big(N(t_k),X(t_k),t_k\big)\Delta t_k\\
&\quad-f\Big(t_k,X(t_k),N(t_k),u\big(N(t_k),X(t_k),t_k\big),\big(\sigma^\top\nabla_x u\big)\big(N(t_k),X(t_k),t_k\big)\Big)\Delta t_k,
\end{aligned}
\tag{2.15}
$$

where $\Delta t_k = t_{k+1}-t_k$, $\Delta B(t_k) = B(t_{k+1})-B(t_k)$, and $N(t_{k+1})-N(t_k)$ satisfies (2.3).

We take the discretized time $\{t_k\}_{0\le k\le N}$, the randomly generated paths $\{N(t_k)\}_{0\le k\le N}$, $\{X(t_k)\}_{0\le k\le N}$, and $\{B(t_k)\}_{0\le k\le N}$ as the input data of the neural network. Letting $\theta = \{\theta_\phi,\theta_{\psi_1},\theta_{\psi_2}\}$, the final output $\hat{u}(\{N(t_k),X(t_k),B(t_k),t_k\}_{0\le k\le N}|\theta)$ is obtained through the scheme (2.15) as an approximation of $g(N(T),X(T))$. The difference from the given terminal condition can be used to construct the loss function

$$
\mathrm{Loss}(\theta)=\mathbb{E}\Big[\big|g\big(N(T),X(T)\big)-\hat{u}\big(\{N(t_k),X(t_k),B(t_k),t_k\}_{0\le k\le N}|\theta\big)\big|^2\Big]. \tag{2.16}
$$

One can refer to Section 5 for more details on the architecture and performance of the neural network. Now, one can optimize the neural network parameters θ using the stochastic gradient descent-type (SGD) algorithm. The numerical experiments in this paper use the Adam optimizer [22].

3 Bivariate kinetic equations for polymer diffusion

In this section, we introduce a class of applications of Eq. (1.1) in polymer dynamics scenarios that accompany chemical reactions [27]. Specifically, we present and simply show the derivation process of the forward and backward Fokker-Planck equations and the Feynman-Kac equations satisfied by the polymer particle CM that accompanies the polymerization and depolymerization reactions through subordination [41].

To describe the diffusion process of polymers, we start with the combined set of stochastic equations

$$
\frac{d}{dt}X(t)=\sqrt{2D\big(N(t)\big)}\xi(t), \tag{3.1a}
$$

$$
D\big(N(t)\big)=\frac{D_0}{\big(N(t)+n_{\min}\big)^\alpha}. \tag{3.1b}
$$

Expression (3.1a) designates the well-known overdamped Langevin equation driven by the white Gaussian noise $\xi(t)$. The diffusion coefficient $D(N(t))$ is a function of the relative size $N(t)$ of the linear polymer particles, which follows polymer physics [8, 10]. In Eq. 3.1b, D_0 is specific to the polymer subunit, α is specific to the selected polymer model, and $n_{\min}$ corresponds to the size of a polymer nucleus.

We consider the birth-death process $N(t)$ of fluctuating relative size of polymer particle described in [27], which is the simplest linear polymer [28]. The process $N(t)$ satisfies $N(t)\in\mathbb{N}$ for $t\ge 0$,

$$
\mathbb{P}\big(N(t+\Delta t)=j|N(t)=i\big)=p_{ij}(\Delta t), \tag{3.2}
$$

$$p_{ij}(\Delta t)=\begin{cases}\lambda(i)\Delta t+o(\Delta t), & j=i+1,\\ \mu(i)\Delta t+o(\Delta t), & j=i-1,\\ 1-\lambda(i)\Delta t-\mu(i)\Delta t+o(\Delta t), & j=i,\\ o(\Delta t), & \text{otherwise}\end{cases} \tag{3.3}$$

for small Δt, where $\lambda(i)$ and $\mu(i)$ are non-negative constants for all i and $\mu(0)=0$.

To apply the theory of subordination [5], we first consider the integral functional

$$\tau(t)=\int_0^t D\big(N(s)\big)ds, \tag{3.4}$$

as well as the joint probability density

$$u(n,\tau,t)=\mathbb{E}\Big[\mathbb{I}_{\{N(t)=n\}}\delta(\tau(t)-\tau)\Big],$$

and the conditional probability density

$$u_{n_0}(\tau,t)=\mathbb{E}\big[\delta\big(\tau(t)-\tau\big)|N(0)=n_0\big],$$

where $\mathbb{I}$ is the indicator function and $\delta(x)$ is the Dirac function.

Through simple derivation (see [41] or Appendices A and B), one can obtain the following two classes of subordinate equations.

Lemma 3.1. *The forward subordinate equations governing the probability density $u(n,\tau,t)$ reads*

$$\begin{cases}\dfrac{\partial}{\partial t}u(n,\tau,t)=\mathcal{L}_n u(n,\tau,t)-D(n)\dfrac{\partial}{\partial\tau}u(n,\tau,t),\\ u(n,\tau,0)=g_n(n)\delta(\tau),\end{cases} \tag{3.5}$$

where

$$\mathcal{L}_n f(n)=\begin{cases}\mu(n+1)f(n+1)+\lambda(n-1)f(n-1)-\big(\mu(n)+\lambda(n)\big)f(n), & n>0,\\ \mu(1)f(1)-\lambda(0)f(0), & n=0\end{cases} \tag{3.6}$$

is a difference operator on variable n, and $g_n(n)$ is the probability distribution of $N(0)$.

Lemma 3.2. *The backward subordinate equations governing the probability density $u_{n_0}(\tau,t)$ reads*

$$\begin{cases}\dfrac{\partial}{\partial t}u_{n_0}(\tau,t)=\mathcal{F}_{n_0}u_{n_0}(\tau,t)-D(n_0)\dfrac{\partial}{\partial\tau}u_{n_0}(\tau,t),\\ u_{n_0}(\tau,0)=\delta(\tau),\end{cases} \tag{3.7}$$

where

$$\mathcal{F}_{n_0}f_{n_0}=\begin{cases}\lambda(n_0)(f_{n_0+1}-f_{n_0})+\mu(n_0)(f_{n_0-1}-f_{n_0}), & n_0>0,\\ \lambda(0)(f_1-f_0), & n_0=0\end{cases} \tag{3.8}$$

is a difference operator on variable n_0.

Remark 3.1. Note that when letting $\alpha(n) = \lambda(n), \beta(n) = \mu(n)$ for $n \geq 0$, the operator $\mathcal{T}_n$ is the same as $\mathcal{F}_n$. Letting $\lambda(-1) = 0$, $\alpha(n) = \mu(n+1)$, $\beta(n) = \lambda(n-1)$ for $n \geq 0$, the operators $\mathcal{T}_n$ and $\mathcal{L}_n$ satisfy

$$\mathcal{L}_n f(n) = \mathcal{T}_n f(n) + c(n) f(n), \tag{3.9}$$

where $c(n) = \alpha(n) + \beta(n) - \alpha(n-1) - \beta(n+1)$.

One can combine the subordinate equations (3.5) and (3.7) to derive the forward and backward Fokker-Planck equations. First it can be noticed that the probability density function

$$\bar{P}(x,t) = \bar{P}\big(x,t|D\big(N(t)\big)\big) = \mathbb{E}\big[\delta\big(X(t)-x\big)|D\big(N(t)\big)\big]$$

fulfills the diffusion equation

$$\frac{\partial}{\partial t}\bar{P}(x,t) = D\big(N(t)\big)\nabla_x^2 \bar{P}(x,t). \tag{3.10}$$

Then, we can rewrite the Langevin equations (3.1) in the subordinated form [5]

$$\frac{d}{d\tau}X(\tau) = \sqrt{2}\xi(\tau), \tag{3.11a}$$

$$\frac{d}{dt}\tau(t) = D\big(N(t)\big). \tag{3.11b}$$

After this, the Green function of the diffusion equation has the form

$$G(x,\tau) = \frac{1}{\sqrt{(4\pi\tau)^d}} \exp\left(-\frac{|x|^2}{4\tau}\right). \tag{3.12}$$

Therefore, if the distribution of the path τ is known, one can get $P(x,t) = \mathbb{E}[\delta(X(t)-x)]$ by averaging the Green's function over τ in the form

$$P(x,t) = \int_0^\infty T(\tau,t) G(x,\tau) d\tau, \tag{3.13}$$

where $T(\tau,t)$ is the probability density function of the process $\tau(t)$. Eq. (3.13) is the well-known subordination formula.

Consider the probability density $\tilde{P}(x,\tau) = \mathbb{E}[\delta(X(\tau)-x)|\tau]$ and the initial value problem

$$\begin{cases} \dfrac{\partial}{\partial \tau}\tilde{P}(x,\tau) = \nabla_x^2 \tilde{P}(x,\tau), \\ \tilde{P}(x,0) = g_x(x), \end{cases} \tag{3.14}$$

where $g_x(x)$ is the distribution of $X(0)$. The solution to problem (3.14) satisfies

$$\tilde{P}(x,\tau) = \int_{\mathbb{R}^d} G(x-y,\tau) g_x(y) dy = \frac{1}{\sqrt{(4\pi\tau)^n}} \int_{\mathbb{R}^d} e^{-\frac{|x-y|^2}{4\tau}} g_x(y) dy. \tag{3.15}$$

We define

$$u(n,x,t)=\mathbb{E}\left[\mathbb{I}_{\{N(t)=n\}}\delta\big(X(t)-x\big)\right]$$

as the joint distribution of the polymer particle size $N(t)$ and CM position $X(t)$ at time t, and

$$u_{n_0}(x,t)=\mathbb{E}\big[\delta\big(X(t)-x\big)|N(0)=n_0\big]$$

the probability density function of the polymer particle CM position $X(t)$ at time t for $N(0)=n_0$. Then, one can get the following two equations (see [41] or Appendix C).

Theorem 3.1. *The joint probability distribution $u(n,x,t)$ satisfies the forward Fokker-Planck equation*

$$\begin{cases}\dfrac{\partial}{\partial t}u(n,x,t)=\mathcal{L}_n u(n,x,t)+D(n)\nabla_x^2 u(n,x,t),\\ u(n,x,0)=g(n,x),\end{cases}\tag{3.16}$$

where $g(n,x)$ is the given initial joint distribution.

Theorem 3.2. *The probability density function $u_{n_0}(x,t)$ satisfies the backward Fokker-Planck equation*

$$\begin{cases}\dfrac{\partial}{\partial t}u_{n_0}(x,t)=\mathcal{F}_{n_0}u_{n_0}(x,t)+D(n_0)\nabla_x^2 u_{n_0}(x,t),\\ u_{n_0}(x,0)=g_{n_0}(x),\end{cases}\tag{3.17}$$

where $g_{n_0}(x)$ is the initial distribution of particles with an initial size $N(0)=n_0$.

The functionals of diffusive motion have many applications [9, 38], and they arise in numerous problems across a variety of scientific fields from condensed matter physics [6, 14, 21], to hydrodynamics [2], meteorology [23], and finance [7, 25]. Consider the position functional

$$Y(t)=\int_0^t U\big(X(\tau)\big)d\tau,\tag{3.18}$$

where $U:\mathbb{R}^d\to\mathbb{R}^1$ is a given function.

We define

$$u(n,x,y,t)=\mathbb{E}\left[\mathbb{I}_{\{N(t)=n\}}\delta\big(X(t)-x\big)\delta\big(Y(t)-y\big)\right]$$

as the joint probability distribution of polymer size $N(t)$, CM position $X(t)$, functional $Y(t)$ at time t, and

$$u_{n_0,x_0}(y,t)=\mathbb{E}\big[\delta\big(Y(t)-y\big)|N(0)=n_0,X(0)=x_0\big]$$

as the probability density function of functional $Y(t)$ at time t given $N(0)=n_0, X(0)=x_0$. Below, we give the forward Feynman-Kac equations governing the joint distribution $u(n,x,y,t)$ and the backward Feynman-Kac equations governing the distribution $u_{n_0,x_0}(y,t)$ (see [41] or Appendix D).

Theorem 3.3. *Let*

$$\widetilde{u}(n,x,p,t)=\int_{\mathbb{R}}u(n,x,y,t)e^{-ipy}dy$$

be the Fourier transform of the joint probability density $u(n,x,y,t)$ with respect to $y\to p$. Then it satisfies the forward bivariate Feynman-Kac equations

$$\begin{cases}\dfrac{\partial}{\partial t}\widetilde{u}(n,x,p,t)=\mathcal{L}_n\widetilde{u}(n,x,p,t)+D(n)\nabla_x^2\widetilde{u}(n,x,p,t)-ipU(x)\widetilde{u}(n,x,p,t),\\ \widetilde{u}(n,x,p,0)=g(n,x).\end{cases}\tag{3.19}$$

Theorem 3.4. *Let*

$$\widetilde{u}_{n_0,x_0}(p,t)=\int_{\mathbb{R}}u_{n_0,x_0}(y,t)e^{-ipy}dy$$

be the Fourier transform of the probability density $u_{n_0,x_0}(y,t)$ with respect to $y\to p$. Then it satisfies the backward Feynman-Kac equations

$$\begin{cases}\dfrac{\partial}{\partial t}\widetilde{u}_{n_0,x_0}(p,t)=\mathcal{F}_{n_0}\widetilde{u}_{n_0,x_0}(p,t)+D(n_0)\nabla_{x_0}^2\widetilde{u}_{n_0,x_0}(p,t)-ipU(x_0)\widetilde{u}_{n_0,x_0}(p,t),\\ \widetilde{u}_{n_0,x_0}(p,0)=1.\end{cases}\tag{3.20}$$

So far, we have obtained the forward and backward Fokker-Planck equations (3.16), (3.17) and the Feynman-Kac equations (3.19), (3.20) for the polymer dynamics accompanying the polymerization and depolymerization reactions. Each of these four types of equations contains a discrete variable n related to the size of the polymer particle, governed by two types of discrete operators (3.6), (3.8). By Remark 3.1, one can easily generalize these four types of equations into the form of Eq. (1.1). As mentioned above, they are infinitely dimensional coupled systems. Next, we will obtain their numerical solutions through the deep BSDE method developed in Section 2.

4 Numerical results

In this section, we will use the deep BSDE method constructed in Section 2 to solve the four types of bivariate kinetic equations of polymers obtained in Section 3 as well as one type of polymer reaction diffusion equation. To test the robustness of our deep BSDE method, we construct various types of discontinuous $\alpha(n)$ and $\beta(n)$ in numerical experiments in this section, which are much more complex than the constant and linear rates [1,27], thus providing a guarantee for practical applications. At the same time, we demonstrate the effectiveness of deep BSDE method by comparing the numerical solution of the neural network with the "exact solution" obtained by Monte-Carlo method (MCM). To apply the deep BSDE method, as described in Section 2, we consider the equations after the time transformation $t\to T-t$ below and keep the notation consistent.

4.1 Backward Fokker-Planck equation

Consider the backward Fokker-Planck equation

$$\begin{cases} \dfrac{\partial}{\partial t}u_{n_0}(x,t) + \mathcal{F}_{n_0}u_{n_0}(x,t) + D(n_0)\nabla_x^2 u_{n_0}(x,t) = 0, \\ u_{n_0}(x,T) = g_{n_0}(x), \end{cases} \tag{4.1}$$

governing the probability density function $u_{n_0}(x,t)$ of polymer CM position $X(t)$ for $N(0) = n$, where

$$\mathcal{F}_{n_0}f_{n_0} = \begin{cases} \lambda(n_0)(f_{n_0+1} - f_{n_0}) + \mu(n_0)(f_{n_0-1} - f_{n_0}), & n_0 \geq 1, \\ \lambda(0)(f_1 - f_0), & n_0 = 0, \end{cases} \tag{4.2}$$

and $\lambda(n_0), \mu(n_0)$ are given functions with respect to n_0. Applying Remark 3.1 and Theorem 2.1, one can get the BSDE

$$\begin{aligned} u_{N(t)}\big(X(t),t\big) = u_{N(0)}\big(X(0),0\big) &+ \int_0^t \sqrt{2D\big(N(\tau)\big)}\nabla_x u_{N(\tau)}\big(X(\tau),\tau\big)\cdot dB(\tau) \\ &+ \int_0^t \int_{\mathbb{Z}\setminus\{0\}} u_{N(\tau-)+n}\big(X(\tau),\tau\big) - u_{N(\tau-)}\big(X(\tau),\tau\big)\widetilde{J}\big(d\tau, dn; N(\tau-)\big). \end{aligned} \tag{4.3}$$

We choose $d = 3, T = 1, D_0 = 5, n_{\min} = 3, \alpha = 1$,

$$\alpha(n) = \lambda(n) = \begin{cases} 20 - n, & 0 \leq n < 5, \\ \sqrt{40 - n}, & 5 \leq n < 10, \\ 30 - n, & 10 \leq n < 15, \\ (100 - n)/10, & 15 \leq n < 100, \\ 0, & 100 \leq n, \end{cases}$$

$$\beta(n) = \mu(n) = \begin{cases} n, & 0 \leq n < 10, \\ n - 5, & 10 \leq n < 20, \\ \sqrt{n}, & 20 \leq n < 30, \\ 5, & 30 \leq n, \end{cases}$$

and the terminal condition

$$g_{n_0}(x) = \begin{cases} \left(\dfrac{10}{\pi(n_0+1)}\right)^{\frac{d}{2}} e^{-\frac{10|x|^2}{n_0+1}}, & n_0 = 3k+1, \\ 0, & \text{otherwise} \end{cases} \tag{4.4}$$

for $(n_0, x) \in \mathbb{N} \times \mathbb{R}^d$. According to Corollary 2.1, it can be obtained that the exact solution of (4.1) has the form

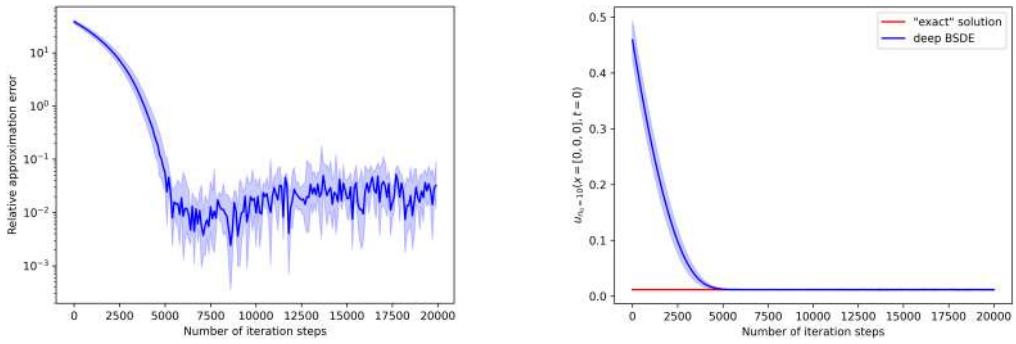
$$u_{n_0}(x,t) = \mathbb{E}\left[\left(\frac{10}{\pi(N_T^{n_0,t}+1)}\right)^{\frac{d}{2}} e^{-\frac{10|X_T^{x,t}|^2}{N_T^{n_0,t}+1}} \mathbb{I}_{\{N_T^{n_0,t} \equiv 1 (\mathrm{mod}\, 3)\}}\right]. \tag{4.5}$$

We get an approximation of $u_{n_0=10}(x=[0,0,0],t=0)$ using the deep BSDE method. Fig. 4.1(a) shows the mean $\pm$ the SD of the relative error for the deep BSDE solutions for five runs, and the "exact solution" $u_{n_0=10}(x=[0,0,0],t=0)\approx 0.011642$ obtained from the MCM of $1,000,000$ random paths over $1,000$ equidistant time steps. The relative error of 2.26% in $20,000$ iteration steps is obtained.

On the other hand, we choose the terminal condition

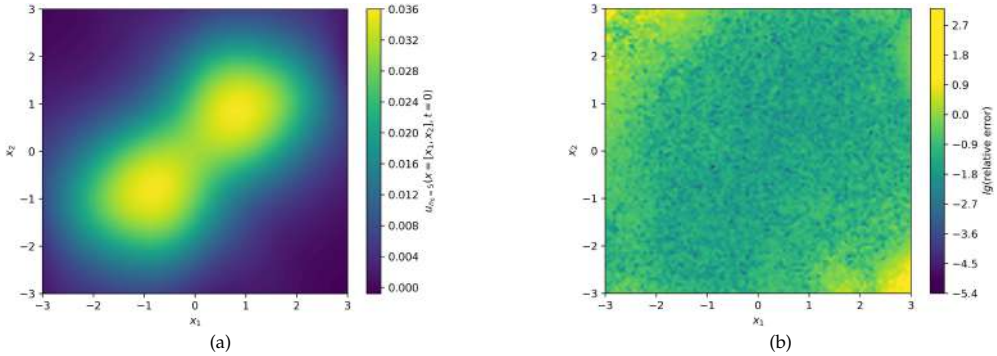
$$g_{n_0}(x)=\begin{cases}\dfrac{1}{2}\left(\dfrac{n_0+1}{\pi}\right)^{\frac{d}{2}}\left(e^{-|x-1|^2(n_0+1)}+e^{-|x+1|^2(n_0+1)}\right), & n_0=2k+1,\\ 0, & \text{otherwise},\end{cases} \tag{4.6}$$

and solve the backward Fokker-Planck equation (4.1) in the 2-dimensional ($d=2$) region $D=[-3,3]\times[-3,3]$. Fig. 4.2 plots the solutions and relative errors. The deep BSDE model



(a) (b)

Figure 4.1: (a) Relative errors with respect to $u_{n_0=10}(x=[0,0,0],t=0)$ for $20,000$ iterations of the deep BSDE method of the backward Fokker-Planck equations (4.1) with 50 equidistant time steps ($N=50$), batch size 128, and learning rate 0.0002. The shaded area depicts the mean $\pm$ the SD of five different runs. (b) The convergence of the deep BSDE solution, where the blue shade represents the mean $\pm$ the SD of five runs, and the red line is for the "exact solution" obtained by MCM.



(a) (b)

Figure 4.2: (a) Plot of $\phi(x|\theta_\phi)$ as an approximation of $u_{n_0=5}(x,t=0)$ for the deep BSDE method. The deep BSDE method goes through $40,000$ iterations of 1024 batch size at 50 equidistant time steps ($N=50$), with a learning rate of 0.001 for the first $20,000$ and 0.0001 for the last $20,000$ iterations. (b) Relative error of the deep BSDE method for $u_{n_0=5}(x,t=0)$, where the "exact solution" is obtained by the MCM.

is trained over $40,000$ iterations, and the "exact solutions" are at a 100×100 uniform grid of points on D, each obtained by the MCM of $10,000$ random samples.

4.2 Forward Fokker-Planck equation

Consider the forward Fokker-Planck equation

$$\begin{cases} \dfrac{\partial}{\partial t}u(n,x,t) + \mathcal{L}_n u(n,x,t) + D(n)\nabla_x^2 u(n,x,t) = 0, \\ u(n,x,T) = g(n,x), \end{cases} \tag{4.7}$$

governing the joint probability distribution $u(n,x,t)$ with respect to the polymer size $N(t)$ and polymer CM position $X(t)$ at time t, where

$$\mathcal{L}_n f(n) = \begin{cases} \mu(n+1)f(n+1) + \lambda(n-1)f(n-1) - \big(\mu(n)+\lambda(n)\big)f(n), & n > 0, \\ \mu(1)f(1) - \lambda(0)f(0), & n = 0, \end{cases} \tag{4.8}$$

and $\mu(n), \lambda(n)$ are given functions with respect to n. Applying Remark 3.1 and Theorem 2.1, one can get the BSDE

$$\begin{aligned} & u\big(N(t),X(t),t\big) - u\big(N(0),X(0),0\big) \\ = & \int_0^t \Big(\alpha\big(N(\tau)-1\big) + \beta\big(N(\tau)+1\big) - \alpha\big(N(\tau)\big) - \beta\big(N(\tau)\big)\Big) u\big(N(\tau),X(\tau),\tau\big)\,d\tau \\ & + \int_0^t \int_{\mathbb{Z}\setminus\{0\}} u\big(N(\tau-)+n,X(\tau),\tau\big) - u\big(N(\tau-),X(\tau),\tau\big)\widetilde{J}\big(d\tau,dn;N(\tau-)\big) \\ & + \int_0^t \sqrt{2D\big(N(\tau)\big)}\nabla_x u\big(N(\tau),X(\tau),\tau\big)\cdot dB(\tau), \end{aligned} \tag{4.9}$$

where $\alpha(n) = \mu(n+1)$ and $\beta(n) = \lambda(n-1)$ with $\lambda(-1) = 0$.

We choose $d = 3, T = 0.5, D_0 = 1, n_{\min} = 3, \alpha = 1$,

$$\lambda(n) = \begin{cases} (20-n)^{\frac{1}{3}}, & 0 \le n < 5, \\ \sqrt{30-n}, & 5 \le n < 10, \\ 20-n, & 10 \le n < 15, \\ (50-n)/10, & 15 \le n < 50, \\ 0, & 50 \le n, \end{cases}$$

$$\mu(n) = \begin{cases} 1.5n, & 0 \le n < 10, \\ n-5, & 10 \le n < 20, \\ \sqrt{n-10}, & 20 \le n < 30, \\ 5, & 30 \le n, \end{cases}$$

and the terminal condition

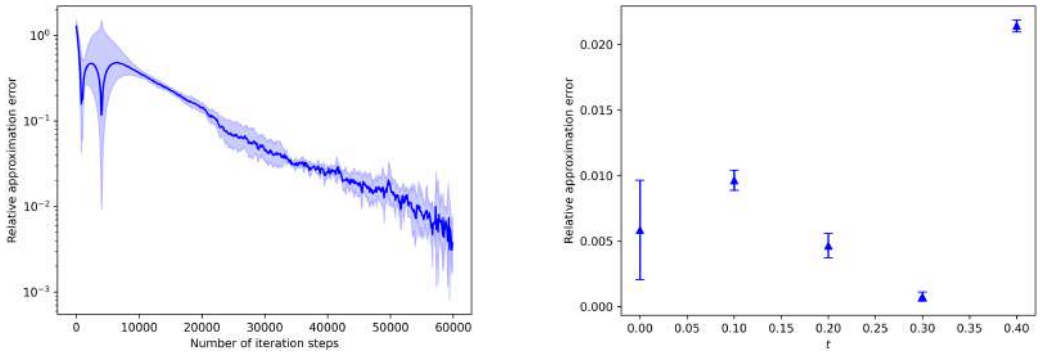
$$g(n,x) = \begin{cases} e^{-\frac{|x|^2}{n+1}}, & n = 2k, \\ 0, & \text{otherwise} \end{cases} \tag{4.10}$$

for $(n,x) \in \mathbb{N} \times \mathbb{R}^d$. According to Corollary 2.1, it can be obtained that the exact solution of (4.5) has the form

$$u(n,x,t) = \mathbb{E}\left[e^{\int_t^T c(N_\tau^{n,t}) d\tau - \frac{|X_T^{x,t}|^2}{N_T^{n,t}+1}} \mathbb{I}_{\{N_T^{n,t} \equiv 0 (\mathrm{mod}\, 2)\}} \right], \tag{4.11}$$

where $c(n) = \alpha(n) + \beta(n) - \alpha(n-1) - \beta(n+1)$.

Fig. 4.3(a) shows the mean $\pm$ the SD of the relative error for the deep BSDE solution $u(n=8, x=[0,0,0], t=0)$. The deep BSDE method achieves a relative error of 0.58% in $60,000$ iteration steps. Fig. 4.3(b) plots the mean $\pm$ the SD of the relative error for the deep BSDE solutions $\{u(n=8, x=[0,0,0], t), t = 0, 0.1, \ldots, 0.4\}$, each obtained by five independent runs, and the "exact solutions" $[0.204230, 0.229959, 0.268836, 0.323641, 0.414984]$ obtained by MCM with $1,000,000$ independent random samples, respectively.



(a) (b)

Figure 4.3: (a) Relative error with respect to $u(n=8, x=[0,0,0], t=0)$ for $60,000$ iterations of the deep BSDE method of the forward Fokker-Planck equation (4.7) with 50 equidistant time steps ($N=50$), batch size 128, and learning rate 0.0004. The shaded area depicts the mean $\pm$ the SD of five different runs. (b) Relative error with respect to $u(n=8, x=[0,0,0], t)$ at five time points $t \in \{0, 0.1, 0.2, 0.3, 0.4\}$ of the forward Fokker-Planck equation (4.7). The results are obtained after $60,000$ iterations of deep BSDE method with 0.01 time step, 128 batch size, and learning rate 0.0004. The blue line represents the mean $\pm$ the SD of five runs.

4.3 Occupation time in half-space

Define the occupation time of a particle in the positive half-space as $Y(t) = \int_0^t U(X(\tau)) d\tau$ ($U(x) = 1$ for $x \geq 0$ and is zero otherwise) [24, 33]. To find the distribution of occupation times, we consider the backward Feynman-Kac equation

$$\begin{cases} \dfrac{\partial}{\partial t} \widetilde{u}_{n_0,x_0}(p,t) + \mathcal{F}_{n_0} \widetilde{u}_{n_0,x_0}(p,t) + D(n) \nabla_{x_0}^2 \widetilde{u}_{n_0,x_0}(p,t) - ipU(x_0) \widetilde{u}_{n_0,x_0}(p,t) = 0, \\ \widetilde{u}_{n_0,x_0}(p,T) = 1, \end{cases} \tag{4.12}$$

governing the function

$$\widetilde{u}_{n_0,x_0}(p,t) = \int_{\mathbb{R}} u_{n_0,x_0}(y,t) e^{-ipy} dy,$$

which is the Fourier transformation of the probability density function $u_{n_0,x_0}(y,t)$ of the polymer CM position functional $Y(t)$ for $N(0)=n_0$ and $X(0)=x_0$, where $\mathcal{F}_{n_0}$ is defined by (4.2). Applying Theorem 2.1, one can get the BSDE

$$\begin{aligned}\widetilde{u}_{N(t),X(t)}(p,t) &= \widetilde{u}_{N(0),X(0)}(p,0) + ip\int_0^t U\big(X(\tau)\big)\widetilde{u}_{N(\tau),X(\tau)}(p,\tau)d\tau \\ &\quad + \int_0^t \int_{\mathbb{Z}\setminus\{0\}} \widetilde{u}_{N(\tau-)+n,X(\tau)}(p,\tau) - \widetilde{u}_{N(\tau),X(\tau)}(p,\tau)\widetilde{J}\big(d\tau,dn;N(\tau-)\big) \\ &\quad + \int_0^t \sqrt{2D\big(N(\tau)\big)}\nabla_{x_0}\widetilde{u}_{N(\tau),X(\tau)}(p,\tau)\cdot dB(\tau). \end{aligned} \tag{4.13}$$

Here, to avoid the influence of the frequency parameter p on the stability of the scheme when solving in the high-frequency region, letting

$$\bar{u}_{N(t),X(t)}(p,t) = e^{-ip\int_0^t U(X(\tau))d\tau}\widetilde{u}_{N(t),X(t)}(p,t),$$

we convert (4.13) to the equivalent BSDE

$$\begin{aligned}&\bar{u}_{N(t),X(t)}(p,t) - \bar{u}_{N(0),X(0)}(p,0) \\ &= \int_0^t e^{-ip\int_0^\tau U(X(s))ds}\int_{\mathbb{Z}\setminus\{0\}} \widetilde{u}_{N(\tau-)+n,X(\tau)}(p,\tau) - \widetilde{u}_{N(\tau),X(\tau)}(p,\tau)\widetilde{J}\big(d\tau,dn;N(\tau-)\big) \\ &\quad + \int_0^t e^{-ip\int_0^\tau U(X(s))ds}\sqrt{2D\big(N(\tau)\big)}\nabla_{x_0}\widetilde{u}_{N(\tau),X(\tau)}(p,\tau)\cdot dB(\tau). \end{aligned} \tag{4.14}$$

Take $d=3, T=1, D_0=3, n_{\min}=4, \alpha=0.5$,

$$\alpha(n)=\lambda(n)=\begin{cases}20-n, & 0\le n<5,\\ \sqrt{40-n}, & 5\le n<10,\\ 30-n, & 10\le n<15,\\ (100-n)/10, & 15\le n<100,\\ 0, & 100\le n,\end{cases}$$

$$\beta(n)=\mu(n)=\begin{cases}n, & 0\le n<10,\\ n-5, & 10\le n<20,\\ \sqrt{n}, & 20\le n<30,\\ 5, & 30\le n,\end{cases}$$

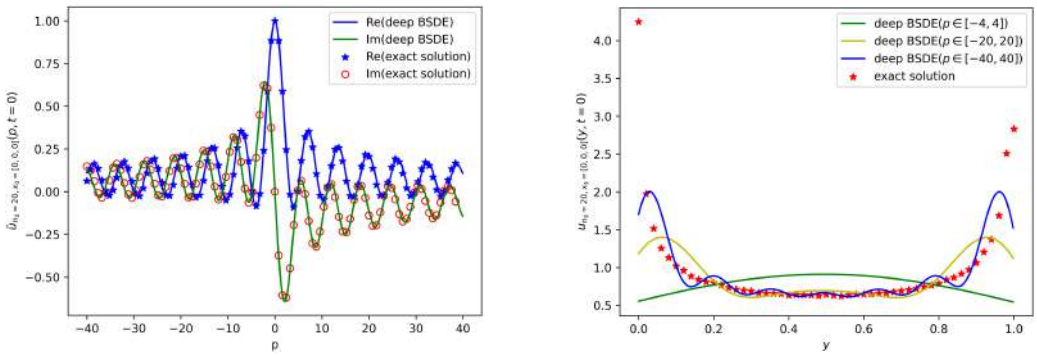
and the terminal condition becomes

$$\bar{u}_{N(t),X(t)}(p,T) = e^{-ip\int_0^T U(X(t))dt}. \tag{4.15}$$

Combining Corollary 2.1 with Remark 3.1, one can get the stochastic representation of (4.12) as

$$\widetilde{u}_{n_0,x_0}(p,t) = \mathbb{E}\left[e^{-ip\int_t^T U(X_\tau^{x,t})d\tau}\right]. \tag{4.16}$$

Fig. 4.4 shows the shape of the deep BSDE solution of the backward Feynman-Kac equation (4.12) in the frequency and time domains, and the shape of the "exact solution" obtained by MCM, where the solution $u_{n_0,x_0}(y,t)$ in the time domain is obtained by taking numerical inverse Fourier transformation $p \to y$ of the deep BSDE solution $\widetilde{u}_{n_0,x_0}(p,t)$ in the frequency domain and the "exact solution" is obtained by simulating $1,000,000$ independent particle trajectories.



(a) (b)

Figure 4.4: (a) The shape of the imaginary and real parts of the numerical solution $\widetilde{u}_{n_0=20,x_0=[0,0,0]}(p,t=0)$ of the deep BSDE method in the interval $[-40,40]$. The results are obtained by 20 independent parallel deep BSDE methods on backward Feynman-Kac equation (4.12) trained on 512 batch size, 100 equidistant time steps ($N=100$), and $40,000$ iterations, with a learning rate of 0.001 for the first $20,000$ and 0.0001 for the last $20,000$. These 20 models are responsible for solving the 20 equidistant interval divisions of the interval $p \in [-40,40]$. The "exact solution" is obtained by approximating the stochastic representation of (4.16) in MCM. (b) The shape of the solutions $u_{n_0=20,x_0=[0,0,0]}(y,t=0)$ in the time domain by taking numerical inverse Fourier transformation $p \to y$ on three different frequency domain intervals $[-40,40]$, $[-20,20]$, and $[-4,4]$ of the deep BSDE method of the backward Feynman-Kac equation (4.12) and the "exact solution" obtained by the MCM.

4.4 The area under the random walk curve

An important application of the functional is the study of NMR [17]. In a typical NMR experiment, the macroscopic measured signal can be written as $E = \mathbb{E}[e^{iY(t)}]$, where $Y(t) = \gamma \int_0^t U(X(\tau))d\tau$ is the phase accumulated by each spin, γ is the gyromagnetic ratio, $U(x)$ is a spatially-inhomogeneous external magnetic field, and $X(\tau)$ is the trajectory of each particle. NMR therefore indirectly encodes information regarding the motion of the particles. Common choices of the magnetic field U are $U(x)=x$ and $U(x)=x^2$ [17].

Consider the forward Feynman-Kac equation

$$\begin{cases} \dfrac{\partial}{\partial t}\widetilde{u}(n,x,p,t) + \mathcal{L}_n\widetilde{u}(n,x,p,t) + D(n)\nabla_x^2\widetilde{u}(n,x,p,t) - ipU(x)\widetilde{u}(n,x,p,t) = 0, \\ \widetilde{u}(n,x,p,T) = g(n,x,p), \end{cases} \tag{4.17}$$

governing the function

$$\widetilde{u}(n,x,p,t) = \int_{\mathbb{R}} u(n,x,y,t)e^{-ipy}dy,$$

which is the Fourier transformation of the joint probability density function $u(n,x,y,t)$ of the polymer particle size $N(t)$, CM position $X(t)$, and functional $Y(t)$, where $\mathcal{L}_n$ is defined by (4.8). Here, we consider a parabolic magnetic field $U(x) = x^2$ and $\gamma = 1$. Applying Theorem 2.1 and Remark 3.1, one can get the BSDE

$$\begin{aligned}
&\widetilde{u}\big(N(t),X(t),p,t\big) - \widetilde{u}\big(N(0),X(0),p,0\big)\\
&= ip\int_0^t U\big(X(\tau)\big)\widetilde{u}\big(N(\tau),X(\tau),p,\tau\big)d\tau\\
&\quad + \int_0^t \int_{\mathbb{Z}\setminus\{0\}} \widetilde{u}\big(N(\tau-)+n,X(\tau),p,\tau\big)\\
&\quad - \widetilde{u}\big(N(\tau-),X(\tau),p,\tau\big)\widetilde{J}\big(d\tau,dn;N(\tau-)\big)\\
&\quad - \int_0^t c\big(N(\tau)\big)\widetilde{u}\big(N(\tau),X(\tau),p,\tau\big)d\tau\\
&\quad + \int_0^t \sqrt{2D\big(N(\tau)\big)}\nabla_x\widetilde{u}\big(N(\tau),X(\tau),p,\tau\big)\cdot dB(\tau),
\end{aligned} \tag{4.18}$$

where $c(n) = \alpha(n) + \beta(n) - \alpha(n-1) - \beta(n+1)$.

We use the same method as in Section 4.3 to avoid instability in high-frequency regions and choose $d = 3, T = 0.5, D_0 = 2, n_{\min} = 3, \alpha = 0.5$,

$$\lambda(n) = \begin{cases} (20-n)^{\frac{1}{3}}, & 0 \le n < 5,\\ \sqrt{30-n}, & 5 \le n < 10,\\ 20-n, & 10 \le n < 15,\\ (50-n)/10, & 15 \le n < 50,\\ 0, & 50 \le n, \end{cases}$$

$$\mu(n) = \begin{cases} 1.5n, & 0 \le n < 10,\\ n-5, & 10 \le n < 20,\\ \sqrt{n-10}, & 20 \le n < 30,\\ 5, & 30 \le n, \end{cases}$$

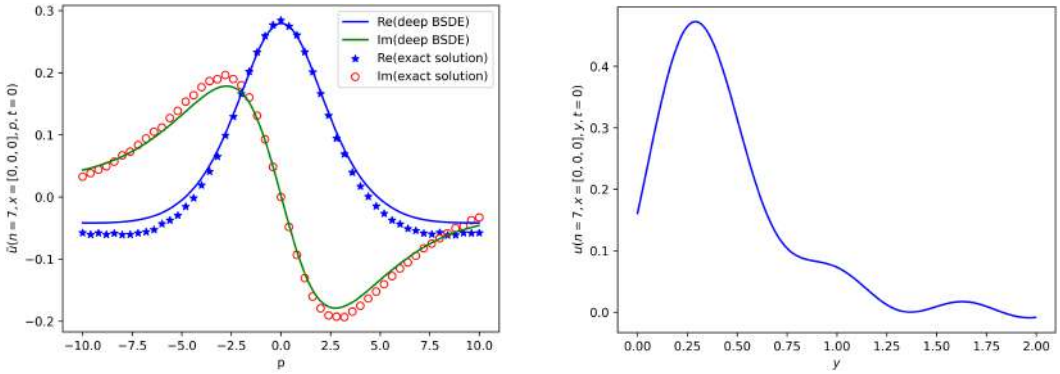
and the terminal condition

$$\begin{aligned}
\bar{u}\big(N(T),X(T),p,T\big) &= e^{-ip\int_0^T U(X(t))dt} g\big(N(T),X(T),p\big)\\
&= \begin{cases} e^{-\frac{|X(T)|^2}{N(T)+1} - ip\int_0^T U(X(t))dt}, & N(T) = 2k+1,\\ 0, & \text{otherwise.} \end{cases}
\end{aligned} \tag{4.19}$$

Similarly, combining Corollary 2.1 with Remark 3.1, one can get the stochastic representation of (4.17) as

$$\widetilde{u}(n,x,p,t) = \mathbb{E}\left[e^{-\frac{|X_T^{x,t}|^2}{N_T^{n,t}+1} + \int_t^T c(N_\tau^{n,t}) - ipU(X_\tau^{x,t})d\tau} \mathbb{I}_{\{N_T^{n,t} \equiv 1 (\mathrm{mod}\, 2)\}} \right]. \tag{4.20}$$

Fig. 4.5 shows the shape of the deep BSDE solution of the forward Feynman-Kac equation (4.17) in the frequency and time domains, where the solution $u(n,x,y,t)$ in the time domain is obtained by taking numerical inverse Fourier transformation $p \to y$ of the solution $\widetilde{u}(n,x,p,t)$ in the frequency domain.



(a) (b)

Figure 4.5: (a) The shape of the imaginary and real parts of the numerical solution $\widetilde{u}(n=7, x=[0,0,0], p, t=0)$ of the deep BSDE method in the interval $[-10,10]$. The results are obtained by deep BSDE method on forward Feynman-Kac equation (4.17) trained on 4096 batch size, 50 equidistant time steps ($N=50$), and $80,000$ iterations, with a learning rate of 0.001 for the first $40,000$ and 0.0005 for the last $40,000$. The "exact solution" is obtained by approximating the stochastic representation of (4.20) in MCM. (b) The shape of the solution $u(n=7, x=[0,0,0], y, t=0)$ in the time domain of the deep BSDE method of the forward Feynman-Kac equation (4.17).

4.5 High-dimensional and nonlinear problems

In order to test our new algorithm's ability to solve high-dimensional nonlinear problems, we consider the high-dimensional Kolmogorov-Petrovsky-Piskunov equation [26] in the polymer diffusion system, and solve this equation using our new algorithm. Consider the nonlinear equation

$$\begin{cases} \dfrac{\partial}{\partial t} u_{n_0}(x,t) + \mathcal{F}_{n_0} u_{n_0}(x,t) + D(n)\nabla_x^2 u_{n_0}(x,t) + u_{n_0}(x,t) - u_{n_0}(x,t)^2 = 0, \\ u_{n_0}(x,T) = g_{n_0}(x). \end{cases} \tag{4.21}$$

We choose $d=100, T=0.5, D_0=5, n_{\min}=3, \alpha=1,$

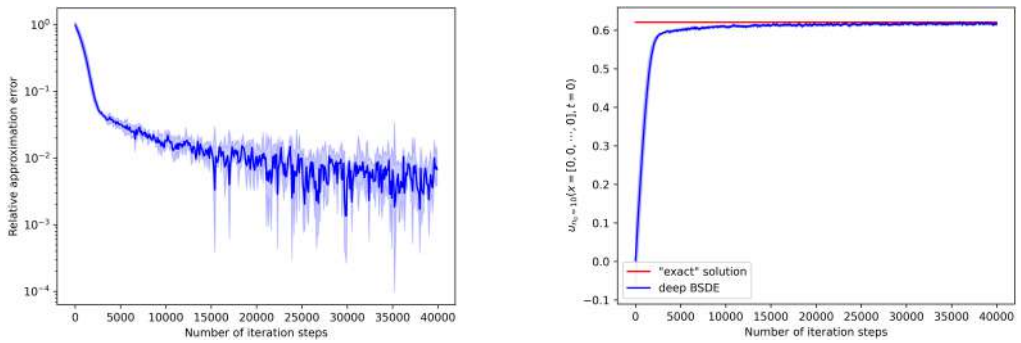
$$\lambda(n) = \begin{cases} 20-n, & 0 \le n < 5, \\ 30-n, & 5 \le n < 10, \\ 40-n, & 10 \le n < 15, \\ 50-n, & 15 \le n < 50, \\ 0, & 50 \le n, \end{cases} \tag{4.22a}$$

$$\mu(n) = \begin{cases} 2n, & 0 \le n < 10, \\ n, & 10 \le n < 20, \\ n/2, & 20 \le n < 30, \\ 7.5, & 30 \le n, \end{cases} \tag{4.22b}$$

and the terminal condition

$$g_{n_0}(x) = \begin{cases} 0.5 - 0.49\left(\sin\left(n_0 \sum_{i=1}^{d} x_i \right) \right), & n_0 = 2k+1, \\ 0.5 - 0.49\left(\cos\left(n_0 \sum_{i=1}^{d} x_i \right) \right), & n_0 = 2k. \end{cases} \tag{4.23}$$

Fig. 4.6 shows the mean $\pm$ the SD of the relative error (a) and the result (b) for the deep BSDE solutions of five independent runs, and the "exact solution" $u_{n_0=10}(x = [0, 0, \cdots, 0], t = 0) \approx 0.620504$ obtained from the branching diffusion method [16] of 100,000 random samples. A relative error of 0.89% is obtained in 40,000 iterations.



(a) (b)

Figure 4.6: (a) Relative error with respect to $u_{n_0=10}(x = [0, 0, \cdots, 0], t = 0)$ for 40,000 iterations of the deep BSDE method of the nonlinear backward Fokker-Planck equations (4.21) with 25 equidistant time steps ($N = 25$), batch size 64, and learning rate 0.0004. The shaded area depicts the mean $\pm$ the SD of five different runs. (b) Plot of the convergence of the deep BSDE method, where the blue shade represents the mean $\pm$ the SD of five runs, and the red line is for the "exact solution" obtained by branching diffusion method [16].

5 Materials and methods

5.1 Neural network architecture

In this subsection, we briefly illustrate the architecture of our new deep BSDE method. Fig. 5.1 shows the architecture of our new deep BSDE method. We use a neural network $\phi(\cdot|\theta_\phi)$ to approximate the target solution u_{t_0} and a neural network $\psi(\cdot|\theta_\psi)$ to the gradient $(\sigma^\top \nabla_x u)(n, x, t)$ and state increments

$$\delta_n^i u(n, x, t) := u(n+i, x, t) - u(n, x, t).$$

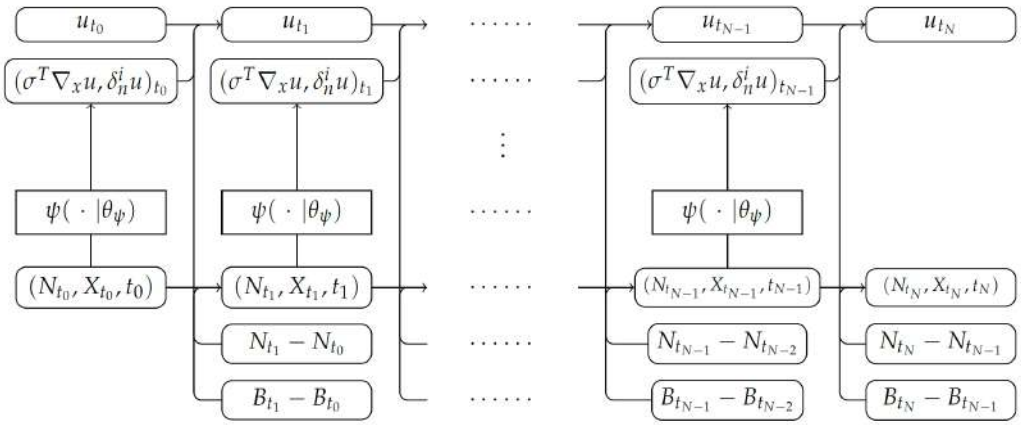


Figure 5.1: The architecture of deep BSDE method used in this paper with N time intervals. u_{t_k} represents the approximate result $u(N_{t_k}, X_{t_k}, t_k)$, where u_{t_0} can be seen as a parameter θ_ϕ and can also be obtained by a neural network $\phi(N_{t_0}, X_{t_0}|\theta_\phi)$ taking $\{N_{t_0}, X_{t_0}\}$ as input, and $\{u_{t_k}\}_{0<k<N}$ is obtained by the explicit discrete scheme (2.15) corresponding to the BSDE (2.9). The neural network $\psi(\cdot\,|\theta_\psi)$ takes (N_{t_k}, X_{t_k}, t_k) as input and outputs the gradient approximation $(\sigma^T\nabla_x u)_{t_k} = (\sigma^T\nabla_x u)(N_{t_k}, X_{t_k}, t_k)$ and the state increments $(\delta_n^i u)_{t_k} = u(N_{t_k} + i, X_{t_k}, t_k) - u(N_{t_k}, X_{t_k}, t_k)$.

To simplify the notations, we denote $N(t)$ as $N_t, X(t)$ as $X_t, u(N_{t_k}, X_{t_k}, t_k)$ as $u_{t_k}, (\sigma^\top\nabla_x u)(N_{t_k}, X_{t_k}, t_k)$ and $\delta_n^i u(N_{t_k}, X_{t_k}, t_k)$ as $(\sigma^\top\nabla_x u)_{t_k}$ and $(\delta_n^i u)_{t_k}$. There are three types of connections in architecture:

i) $(N_{t_k}, X_{t_k}, t_k) \to \psi(\cdot\,|\theta_\psi) \to (\sigma^\top\nabla_x u, \delta_n^i u)_{t_k}$ is a neural network that approximates the gradient term $(\sigma^\top\nabla_x u)(N_{t_k}, X_{t_k}, t_k)$ and state increments $\delta_n^i u(N_{t_k}, X_{t_k}, t_k)$ as described in (2.12) and (2.13). Its parameters θ_ψ and θ_ϕ for approximating $u(N_{t_0}, X_{t_0}, t_0)$ are the goals we need to optimize.

ii) $(u_{t_k}, (\sigma^\top\nabla_x u)_{t_k}, B_{t_{k+1}} - B_{t_k}) \to u_{t_{k+1}}$ is the forward iteration, and the final output is an approximation of $u(N_T, X_T, T)$, which is characterized by the scheme (2.15) corresponding to the BSDE (2.9). There are no parameters to optimize in this type of connection.

iii) $((N_{t_k}, X_{t_k}, t_k), N_{t_{k+1}} - N_{t_k}, B_{t_{k+1}} - B_{t_k}) \to (N_{t_{k+1}}, X_{t_{k+1}}, t_{k+1})$ is the forward propagation process of discretized stochastic processes N_t and X_t accompanied by the update of time t, which is characterized by the definition of stochastic processes and their discrete schemes. This type of connection also has no parameters to optimize.

Here, $\phi(\cdot\,|\theta_\phi)$ and $\psi(\cdot\,|\theta_\psi) = \{\psi_1(\cdot\,|\theta_{\psi_1}), \psi_2(\cdot\,|\theta_{\psi_2})\}$ can be designed as suitable neural networks according to different application scenarios. We map the state (position) information n to d_{model}-dimensional features $\{PE_k(n)\}_k$ through positional encoding method [35], where

$$PE_{2i}(n) = \sin\left(n/10000^{\frac{2i}{d_{model}}}\right) \tag{5.1a}$$

$$PE_{2i+1}(n) = \cos\left(n/10000^{\frac{2i}{d_{model}}}\right), \tag{5.1b}$$

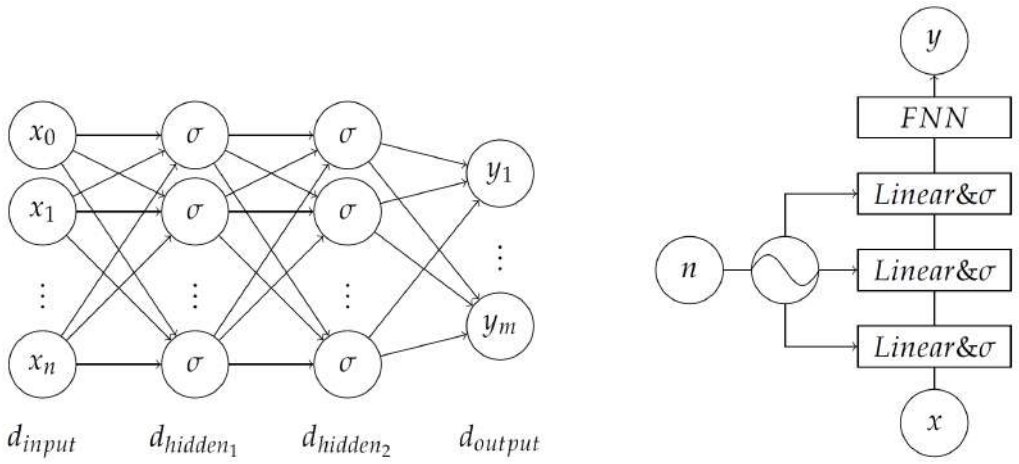
and add these high-dimensional features to the network. We choose the fully connected neural network (FNN) with positional encoding as the approximator and Tanh (5.2) as the activation function

$$\mathrm{Tanh}(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}. \tag{5.2}$$

The network structure is shown in Fig. 5.2. We approximate $(n,x) \mapsto (\sigma^\top \nabla_x u)(n,x,t)$ and $(n,x) \mapsto \delta_n^{\pm} u(n,x,t) := [\delta_n^1 u(n,x,t), \delta_n^{-1} u(n,x,t)]^\top$ at each time step t_k by subnetworks

$$\begin{aligned} &\left(\sigma^\top \nabla_x u\right)(N_{t_k}, X_{t_k}, t_k) \approx \psi_1\left(N_{t_k}, X_{t_k} | \theta_{\psi_1}^k\right) \\ &\delta_n^{\pm} u(N_{t_k}, X_{t_k}, t_k) \approx \psi_2\left(N_{t_k}, X_{t_k} | \theta_{\psi_2}^k\right). \end{aligned} \tag{5.3}$$

We use PyTorch [31] to implement our algorithms and open our source code for readers to test: `https://github.com/WANGH950/deepBSDEmethod`.



(a) Fully connected neural network architecture (b) FNN structure with positional encoding

Figure 5.2: Neural network architecture.

5.2 Approximation effect of neural network

The accuracy of the deep BSDE method depends on the number of hidden layers. To test this factor, we solve the reaction diffusion equation introduced in Section 4.5. We increase the number of hidden layers from 0 to 4 with state information n and report our results in Table 5.1. The results show that the accuracy of the deep BSDE method is improved with the increase of the number of hidden layers of the subnetwork.

We randomly generate 20 8-th-order polynomials $f_n(x)$ to test the approximation ability of FNN with positional encoding structure. Fig. 5.3 shows the exact values of the polynomials and the approximation results of the neural network for $x \in [-1,1]$. We

Table 5.1: The mean and SD of the relative error for the PDE in Eq. 4.21, obtained by the deep BSDE method using FNN with positional with positional encoding of different layers as sub networks.

	No. of hidden layers					
Relative err	0	1	2	3	4	5
Mean,%	2.25	1.27	0.75	0.63	0.56	0.56
SD	0.0017	0.0032	0.0039	0.0032	0.0044	0.0024

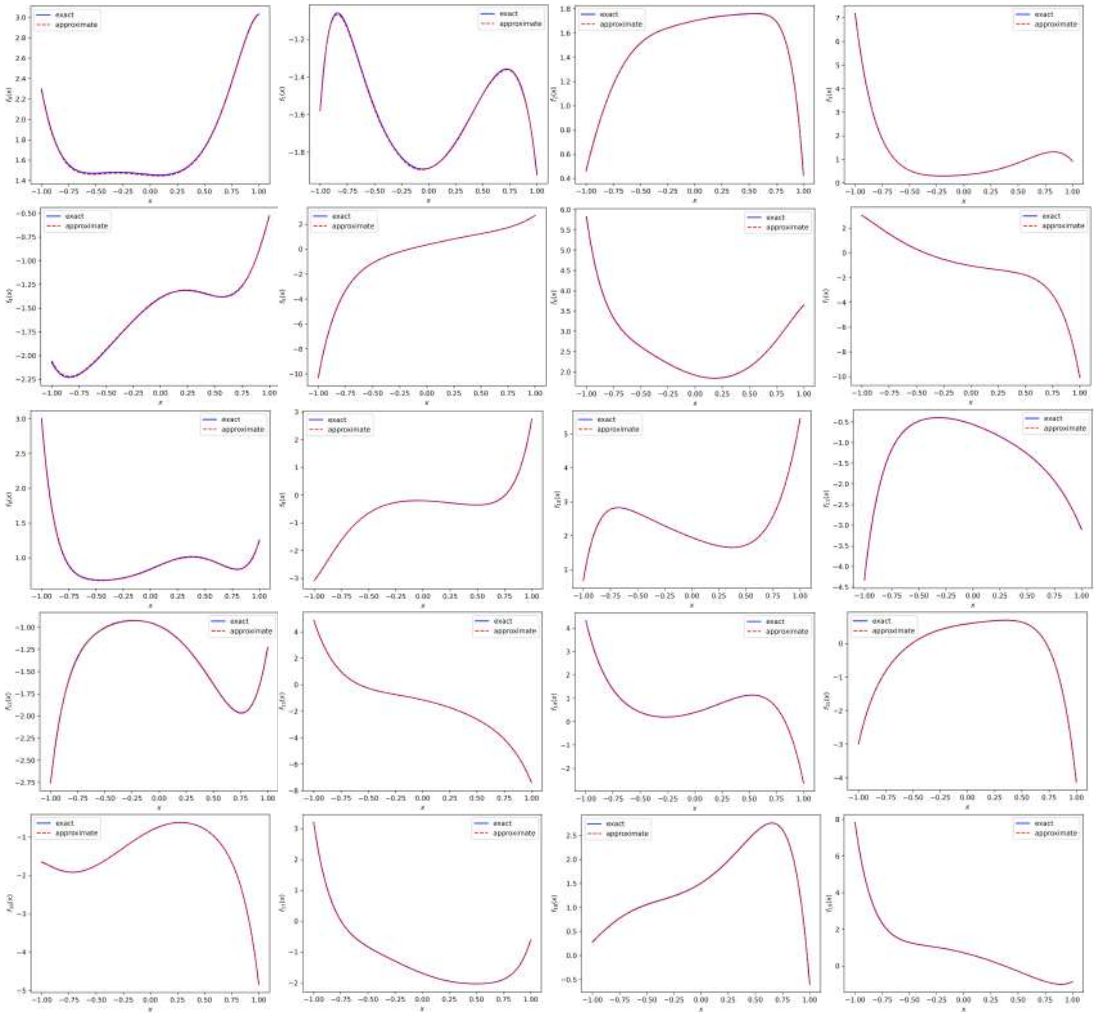


Figure 5.3: FNN with positional encoding structure approximation results for 20 8-th-order polynomials.

use a 3-layer, 128-width FNN with positional coding to approximate polynomials $f_n(x) \approx f_n(x|\theta)$, and the values of each polynomial at 100 equidistant grid points are used as the training set, and the results are obtained after $20,000$ iterations with a learning rate of 0.0001.

6 Discussion

In this paper, we introduce a class of polymer dynamics accompanied by polymerization and depolymerization reactions, and derive the corresponding Fokker-Planck and Feynman-Kac equations, which are a class of infinite-dimensionally coupled systems. Further, we extend the deep BSDE method to solve such equations and demonstrate the effectiveness of the algorithms through extensive numerical experiments.

In reality, there are many cases in polymer systems that need to be modeled by composite stochastic processes [8,10], and we have only discussed one relatively simple linear case. We believe that our method can also handle different types of linear coupling equations by constructing the corresponding process $N(t)$.

For the model in this paper, we have not yet given a complete analysis of the deep BSDE method. It is not difficult to obtain that the deep BSDE method in this paper essentially transforms the solution of PDE (1.1) into the solution of its corresponding forward-backward stochastic differential equations (FBSDEs)

$$\begin{cases} X(t) = X(0) + \int_0^t \mu\big(N(s), X(s), s\big)ds + \int_0^t \sigma\big(N(s), X(s), s\big)dB(s), \\ Y(t) = Y(T) + \int_t^T f\big(s, X(s), N(s), Y(s), Z(s)\big)ds - \int_t^T Z(s)^\top dB(s) \\ \qquad\quad - \int_t^T \int_{\mathbb{Z}\setminus\{0\}} U(s,n)\widetilde{J}\big(ds, dn; N(s-)\big), \end{cases} \tag{6.1}$$

where $\{Y(s), Z(s), U(s,n)\}$ is unknown stochastic processes. Therefore, in order to obtain a complete deep BSDE method, we need to theoretically analyze the properties of FBSDEs (6.1). In addition, we also need to conduct numerical analysis:

(1) Prove that the functions that need to be approximated using neural networks belong to the right function class and bound their norms in that function class.

(2) Estimate the time discretization error.

(3) Adapt the analysis of supervised learning problems applicable to the current setting and give the convergence rate estimate of the numerical solutions [11,15,19].

All of these are also what we intend to study further in the future.

Appendix A. Proof of Lemma 3.1

Proof. Assuming that the polymer has only one polymerization (or depolymerization) reaction in the time interval $[t, t+\Delta t]$, $\Delta t \ll 1$, it can be divided into two cases. When $n > 0$, we have

$$\begin{aligned} u(n,s,t+\Delta t) = {} & \mu_{n+1}\Delta t u\big(n+1, s-\Delta t D(n+1), t\big) \\ & + \lambda_{n-1}\Delta t u\big(n-1, s-\Delta t D(n-1), t\big) \\ & + (1-\mu_n\Delta t - \lambda_n\Delta t)u\big(n, s-\Delta t D(n), t\big) + o(\Delta t). \end{aligned} \tag{A.1}$$

Doing Fourier transform $s \to p$ on both sides of (A.1) leads to

$$\begin{aligned}\widetilde{u}(n,p,t+\Delta t) &= \mu_{n+1}\Delta t e^{ip\Delta tD(n+1)}\widetilde{u}(n+1,p,t) + \lambda_{n-1}\Delta t e^{ip\Delta tD(n-1)}\widetilde{u}(n-1,p,t)\\ &\quad + (1-\mu_n\Delta t - \lambda_n\Delta t)e^{ip\Delta tD(n)}\widetilde{u}(n,p,t) + o(\Delta t)\\ &= \mu_{n+1}\Delta t\widetilde{u}(n+1,p,t) + \lambda_{n-1}\Delta t\widetilde{u}(n-1,p,t) - (\mu_n+\lambda_n)\Delta t\widetilde{u}(n,p,t)\\ &\quad + \widetilde{u}(n,s,t) + ipD(n)\Delta t\widetilde{u}(n,p,t) + o(\Delta t). \end{aligned} \tag{A.2}$$

Therefore,

$$\begin{aligned}&\frac{1}{\Delta t}\big(\widetilde{u}(n,p,t+\Delta t) - \widetilde{u}(n,p,t)\big)\\ &= \mu_{n+1}\widetilde{u}(n+1,p,t) + \lambda_{n-1}\widetilde{u}(n-1,p,t)\\ &\quad - (\mu_n+\lambda_n)\widetilde{u}(n,p,t) + ipD(n)\widetilde{u}(n,p,t) + o(\Delta t). \end{aligned} \tag{A.3}$$

Letting $\Delta t \to 0$, we have

$$\frac{\partial}{\partial t}\widetilde{u}(n,p,t) = \mathcal{L}_n\widetilde{u}(n,p,t) + ipD(n)\widetilde{u}(n,p,t). \tag{A.4}$$

We do the inverse Fourier transform $p \to \tau$ and get forward equation (3.5). When $N(t)$=0, the polymer can only undergo polymerization. Therefore, after the same derivation, the case of $n = 0$ can be proved. The proof is complete. □

Appendix B. Proof of Lemma 3.2

Proof. Assuming that the polymer has at most one polymerization (or depolymerization) process in the time interval $[0, t+\Delta t], \Delta t \ll 1$, it can also be divided into two cases. When $n > 0$, we have

$$\begin{aligned}u_{n_0}(s,t+\Delta t) &= \mu_{n_0}\Delta t u_{n_0-1}\big(s-\Delta tD(n_0),t\big) + \lambda_{n_0}\Delta t u_{n_0+1}\big(s-\Delta tD(n_0),t\big)\\ &\quad + (1-\mu_{n_0}\Delta t - \lambda_{n_0}\Delta t)u_{n_0}\big(s-\Delta tD(n_0),t\big) + o(\Delta t). \end{aligned} \tag{B.1}$$

Doing Fourier transform $s \to p$ on both sides of Eq. B.1 results in

$$\begin{aligned}\widetilde{u}_{n_0}(p,t+\Delta t) &= \mu_{n_0}\Delta t e^{ip\Delta tD(n_0)}\widetilde{u}_{n_0-1}(p,t) + \lambda_{n_0}\Delta t e^{ip\Delta tD(n_0)}\widetilde{u}_{n_0+1}(p,t)\\ &\quad + (1-\mu_{n_0}\Delta t - \lambda_{n_0}\Delta t)e^{ip\Delta tD(n_0)}\widetilde{u}_{n_0}(p,t) + o(\Delta t)\\ &= \mu_{n_0}\Delta t\widetilde{u}_{n_0-1}(p,t) + \lambda_{n_0}\Delta t\widetilde{u}_{n_0+1}(p,t) - (\mu_{n_0}+\lambda_{n_0})\Delta t\widetilde{u}_{n_0}(p,t)\\ &\quad + \widetilde{u}_{n_0}(p,t) + ipD(n_0)\Delta t\widetilde{u}_{n_0}(p,t) + o(\Delta t). \end{aligned} \tag{B.2}$$

Therefore,

$$\begin{aligned}&\frac{1}{\Delta t}\big(\widetilde{u}_{n_0}(p,t+\Delta t) - \widetilde{u}_{n_0}(p,t)\big)\\ &= \mu_{n_0}\widetilde{u}_{n_0-1}(p,t) + \lambda_{n_0}\widetilde{u}_{n_0+1}(p,t)\end{aligned}$$

$$-(\mu_{n_0}+\lambda_{n_0})\widetilde{u}_{n_0}(p,t)+ipD(n_0)\Delta t\widetilde{u}_{n_0}(p,t)+o(\Delta t). \tag{B.3}$$

Letting $\Delta t\to 0$, we have

$$\frac{\partial}{\partial t}\widetilde{u}_{n_0}(p,t)=\mathcal{F}_{n_0}\widetilde{u}_{n_0}(p,t)+ipD(n_0)\widetilde{u}_{n_0}(p,t). \tag{B.4}$$

We do the inverse Fourier transform $p\to\tau$ and get backward equation (3.7). When $N(0)=0$, the polymer can only undergo polymerization. Therefore, after the same derivation, the case of $n_0=0$ can be proved. The proof is complete. □

Appendix C. Proof of Theorems 3.1 and 3.2

Proof of Theorem 3.1. With the help of (3.13), using Eq. 3.5 leads to

$$\begin{aligned}
\frac{\partial}{\partial t}u(n,x,t)&=\frac{\partial}{\partial t}\int_0^\infty \tilde{P}(x,\tau)u(n,\tau,t)d\tau\\
&=\int_0^\infty \tilde{P}(x,\tau)\frac{\partial}{\partial t}u(n,\tau,t)d\tau\\
&=\int_0^\infty \tilde{P}(x,\tau)\left[\mathcal{L}_n u(n,\tau,t)-D(n)\frac{\partial}{\partial\tau}u(n,\tau,t)\right]d\tau\\
&=\mathcal{L}_n f(n,x,t)-D(n)\int_0^\infty \tilde{P}(x,\tau)\frac{\partial}{\partial\tau}u(n,\tau,t)d\tau\\
&=\mathcal{L}_n f(n,x,t)-D(n)\left[\tilde{P}(x,\tau)u(n,\tau,t)|_{\tau=0}^{\tau=\infty}-\int_0^\infty u(n,\tau,t)\frac{\partial}{\partial\tau}\tilde{P}(x,\tau)d\tau\right].
\end{aligned}$$

With the relation $\partial\tilde{P}(x,\tau)/\partial\tau=\nabla_x^2\tilde{P}(x,\tau)$, we get Eq. (3.16). □

Proof of Theorem 3.2. The proof uses Lemma 3.2 and Eq. 3.13, and the procedure of proof is similar to that of Theorem 3.1, so it is omitted. □

Appendix D. Proof of Theorems 3.3 and 3.4

Proof of Theorem 3.3. Respectively taking Fourier transform to $u(n,x,y,t)$ with $n\to l$, $x\to k$, $y\to p$, we have

$$\widetilde{u}(l,k,p,t)=\mathbb{E}\left[e^{-ilN(t)}e^{-ikX(t)}e^{-ipY(t)}\right].$$

The difference of the joint PDF at different times in Fourier space satisfies

$$\begin{aligned}
&\widetilde{u}(l,k,p,t+\Delta t)-\widetilde{u}(l,k,p,t)\\
&=\mathbb{E}\left[e^{-ilN(t+\Delta t)}e^{-ikX(t+\Delta t)}e^{-ipY(t+\Delta t)}\right]-\mathbb{E}\left[e^{-ilN(t)}e^{-ikX(t)}e^{-ipY(t)}\right]\\
&=\mathbb{E}\left[e^{-ilN(t+\Delta t)}e^{-ik(X(t)+\sqrt{2D(N(t))}(B(t+\Delta t)-B(t)))}e^{-ip(Y(t)+U(X(t))\Delta t)}\right]
\end{aligned}$$

$$
\begin{aligned}
&- \mathbb{E}\Big[e^{-ilN(t)}e^{-ikX(t)}e^{-ipY(t)}\Big] \\
&= \Delta t\mathbb{E}\Big[\lambda\big(N(t)\big)e^{-i(N(t)+1)}e^{-ik(X(t)+\sqrt{2D(N(t))}(B(t+\Delta t)-B(t)))}e^{-ip(Y(t)+U(X(t))\Delta t)}\Big] \\
&\quad + \Delta t\mathbb{E}\Big[\mu\big(N(t)\big)e^{-i(N(t)-1)}e^{-ik(X(t)+\sqrt{2D(N(t))}(B(t+\Delta t)-B(t)))}e^{-ip(Y(t)+U(X(t))\Delta t)}\Big] \\
&\quad + \Big[\big(1-\lambda\big(N(t)\big)\Delta t-\mu\big(N(t)\big)\Delta t\big)e^{-i(N(t))}e^{-ik(X(t)+\sqrt{2D(N(t))}(B(t+\Delta t)-B(t)))} \\
&\qquad \times e^{-ip(Y(t)+U(X(t))\Delta t)}\Big] - \mathbb{E}\Big[e^{-ilN(t)}e^{-ikX(t)}e^{-ipY(t)}\Big],
\end{aligned}
$$

where we have used the definitions of $N(t)$, $X(t)$, and $Y(t)$.

Noting that

$$
\begin{aligned}
&e^{-ik\sqrt{2D(N(t))}(B(t+\Delta t)-B(t))} = 1-ik\sqrt{2D\big(N(t)\big)}\big(B(t+\Delta t)-B(t)\big)-k^2D\big(N(t)\big)\Delta t+o(\Delta t), \\
&e^{-ipU(X(t))\Delta t} = 1-ipU\big(X(t)\big)\Delta t+o(\Delta t),
\end{aligned}
$$

we have

$$
\begin{aligned}
&\widetilde{u}(l,k,p,t+\Delta t)-\widetilde{u}(l,k,p,t) \\
&= \Delta t\mathbb{E}\Big[\lambda(N(t))e^{-i(N(t)+1)}e^{-ikX(t)}e^{-ipY(t)}\Big] \\
&\quad + \Delta t\mathbb{E}\Big[\mu\big(N(t)\big)e^{-i(N(t)-1)}e^{-ikX(t)}e^{-ipY(t)}\Big] \\
&\quad - k^2\Delta t\mathbb{E}\Big[D\big(N(t)\big)e^{-i(N(t))}e^{-ikX(t)}e^{-ipY(t)}\Big] \\
&\quad - ip\Delta t\mathbb{E}\Big[U\big(X(t)\big)e^{-i(N(t))}e^{-ikX(t)}e^{-ipY(t)}\Big] \\
&\quad - \Delta t\mathbb{E}\Big[\big(\lambda(N(t))+\mu\big(N(t)\big)\big)e^{-iN(t)}e^{-ikX(t)}e^{-ipY(t)}\Big]+o(\Delta t).
\end{aligned}
$$

Dividing both sides by Δt, taking the inverse Fourier transform with $l \to n, k \to x$, and letting $\Delta t \to 0$, we can get the forward Feynman-Kac equation (3.19), where the derivation of the $n = 0$ case is similar and omitted. □

Proof of Theorem 3.4. Taking Fourier transform to $u_{n_0,x_0}(y,t)$ with $y \to p$, we have

$$
\widetilde{u}_{n_0,x_0}(p,t) = \mathbb{E}\Big[e^{-ipY(t)|_{n_0,x_0}}\Big].
$$

The difference of the PDF at different times in Fourier space satisfies

$$
\begin{aligned}
&\widetilde{u}_{n_0,x_0}(p,t+\Delta t)-\widetilde{u}_{n_0,x_0}(p,t) \\
&= \mathbb{E}\Big[e^{-ipY(t+\Delta t)|_{n_0,x_0}}\Big]-\mathbb{E}\Big[e^{-ipY(t)|_{n_0,x_0}}\Big] \\
&= \mathbb{E}\Big[e^{-ip(U(x_0)\Delta t+Y(t)|_{N(\Delta t),X(\Delta t))}}\Big]-\mathbb{E}\Big[e^{-ipY(t)|_{n_0,x_0}}\Big] \\
&= \lambda(n_0)\Delta t\mathbb{E}\Big[e^{-ipU(x_0)\Delta t}e^{-ipY(t)|_{n_0+1,x_0+\sqrt{2D(n_0)}B(\Delta t)}}\Big]
\end{aligned}
$$

$$
\begin{aligned}
&+ \mu(n_0)\Delta t\mathbb{E}\left[e^{-ipU(x_0)\Delta t}e^{-ipY(t)|_{n_0-1,x_0+\sqrt{2D(n_0)}B(\Delta t)}}\right] \\
&+ \big(1-\lambda(n_0)\Delta t-\mu(n_0)\Delta t\big)\mathbb{E}\left[e^{-ipU(x_0)\Delta t}e^{-ipY(t)|_{n_0,x_0+\sqrt{2D(n_0)}B(\Delta t)}}\right] \\
&- \mathbb{E}\left[e^{-ipY(t)|_{n_0,x_0}}\right].
\end{aligned}
$$

Taking Fourier transform on both sides with $x_0 \to k_0$ and omitting the higher-order terms of Δt, there exists

$$
\begin{aligned}
\widetilde{u}_{n_0,k_0}(p,t+\Delta t)-\widetilde{u}_{n_0,k_0}(p,t) \approx\ & \mu(n_0)\Delta t\widetilde{u}_{n_0-1,k_0}(p,t)+\lambda(n_0)\Delta t\widetilde{u}_{n_0+1,k_0}(p,t) \\
& -\big(\mu(n_0)+\lambda(n_0)\big)\Delta t\widetilde{u}_{n_0,k_0}(p,t)-k_0^2D(n_0)\Delta t\widetilde{u}_{n_0,k_0}(p,t) \\
& -ip\Delta t\mathcal{F}_{x_0\to k_0}\{U(x_0)\widetilde{u}_{n_0,x_0}(p,t)\}.
\end{aligned}
$$

Dividing both sides by Δt, taking the inverse Fourier transform with $k_0 \to x_0$, and letting $\Delta t \to 0$, one can get the backward Feynman-Kac equation (3.20), where the derivation of the $n_0 = 0$ case is similar and omitted. □

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant Nos. 12225107, 12071195), by the Major Science and Technology Projects in Gansu Province-Leading Talents in Science and Technology (Grant No. 23ZDKA0005), by the Innovative Groups of Basic Research in Gansu Province (Grant No. 22JR5RA391), and by the Supercomputing Center of Lanzhou University.

References

[1] F. Baldovin, E. Orlandini, and F. Seno, Polymerization induces non-Gaussian diffusion, *Front. Phys.*, 7:124, 2019.

[2] A. Baule and R. Friedrich, Investigation of a generalized Obukhov model for turbulence, *Phys. Lett. A*, 350(3):167–173, 2006.

[3] S. Bochner, *Harmonic Analysis and the Theory of Probability*, in: *Dover Books on Mathematics*, Dover Publications, 2005.

[4] I. Chakraborty and Y. Roichman, Disorder-induced Fickian, yet non-Gaussian diffusion in heterogeneous media, *Phys. Rev. Research*, 2(2):022020, 2020.

[5] A. V. Chechkin, F. Seno, R. Metzler, and I. M. Sokolov, Brownian yet non-Gaussian diffusion: From superstatistics to subordination of diffusing diffusivities, *Phys. Rev. X*, 7(2):021002, 2017.

[6] A. Comtet, J. Desbois, and C. Texier, Functionals of Brownian motion, localization and metric graphs, *J. Phys. A*, 38(37):R341, 2005.

[7] A. Comtet, C. Monthus, and M. Yor, Exponential functionals of Brownian motion and disordered systems, *J. Appl. Probab.*, 35(2):255–271, 1998.

[8] P.-G. de Gennes, *Scaling Concepts in Polymer Physics*, Cornell University Press, 1979.

[9] W. H. Deng, X. C. Wu, and W. L. Wang, Mean exit time and escape probability for the anomalous processes with the tempered power-law waiting times, *Europhysics Letters*, 117(1):10009, 2017.

[10] M. Doi and S. F. Edwards, *The Theory of Polymer Dynamics*, in: *International Series of Monographs on Physics*, Oxford University Press, 1988.

[11] W. E, J. Han, and A. Jentzen, Algorithms for solving high dimensional PDEs: From nonlinear Monte Carlo to machine learning, *Nonlinearity*, 35(1):278–310, 2021.
[12] W. E and B. Yu, The deep Ritz method: A deep learning-based numerical algorithm for solving variational problems, *Commun. Math. Stat.*, 6:1–12, 2018.
[13] W. Feller, *An Introduction to Probability Theory and Its Application. Volume 2*, in: *Wiley Series in Probaility and Mathematical Statistics*, John Wiley & Sons, 1971.
[14] G. Foltin, K. Oerding, Z. Rácz, R. L. Workman, and R. K. P. Zia, Width distribution for random-walk interfaces, *Phys. Rev. E*, 50:R639–R642, 1994.
[15] C. Gao, S. Gao, R. Hu, and Z. Zhu, Convergence of the backward deep BSDE method with applications to optimal stopping problems, *SIAM J. Financ. Math.*, 14(4):1290–1303, 2023.
[16] E. Gobet, *Monte-Carlo Methods and Stochastic Processes: From Linear to Non-Linear*, Chapman & Hall/CRC, 2016.
[17] D. S. Grebenkov, NMR survey of reflected Brownian motion, *Rev. Mod. Phys.*, 79:1077–1137, 2007.
[18] J. Han, A. Jentzen, and W. E, Solving high-dimensional partial differential equations using deep learning, *Proc. Natl. Acad. Sci. USA*, 115(34):8505–8510, 2018.
[19] J. Han and J. Long, Convergence of the deep BSDE method for coupled FBSDEs, *Probab. Uncertain. Quant. Risk*, 5:5, 2020.
[20] S. Hapca, J. W. Crawford, and I. M. Young, Anomalous diffusion of heterogeneous populations characterized by normal diffusion at the individual level, *J. R. Soc. Interface*, 6(30):111–122, 2009.
[21] G. Hummer and A. Szabo, Free energy reconstruction from nonequilibrium single-molecule pulling experiments, *Proc. Natl. Acad. Sci. USA*, 98(7):3658–3661, 2001.
[22] D. P. Kingma and J. Ba, Adam: A method for stochastic optimization, *arXiv:1412.6980*, 2014.
[23] S. N. Majumdar and A. J. Bray, Large-deviation functions for nonlinear functionals of a Gaussian stationary Markov process, *Phys. Rev. E*, 65:051112, 2002.
[24] S. N. Majumdar and A. Comtet, Local and occupation time of a particle diffusing in a random medium, *Phys. Rev. Lett.*, 89:060601, 2002.
[25] Y. Marc, *Exponential Functionals of Brownian Motion and Related Processes*, Springer, 2001.
[26] H. P. McKean, Application of Brownian motion to the equation of Kolmogorov-Petrovskii-Piskunov, *Commun. Pure Appl. Math.*, 28:323–331, 1975.
[27] S. Nampoothiri, E. Orlandini, F. Seno, and F. Baldovin, Brownian non-Gaussian polymer diffusion and queuing theory in the mean-field limit, *New J. Phys.*, 24(2):023003, 2022.
[28] G. Odian, *Principles of Polymerization*, Wiley-Interscience, 2004.
[29] E. Pardoux and S. Peng, Backward stochastic differential equations and quasilinear parabolic partial differential equations, in: *Stochastic Partial Differential Equations and Their Applications*, Springer, 200–217, 1992.
[30] E. Pardoux and S. Tang, Forward-backward stochastic differential equations and quasilinear parabolic PDEs, *Probab. Theory Relat. Fields*, 114:123–150, 1999.
[31] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, *PyTorch: An Imperative Style, High-Performance Deep Learning Library*, Curran Associates Inc., 2019.
[32] 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.
[33] S. Sabhapandit, S. N. Majumdar, and A. Comtet, Statistical properties of functionals of the paths of a particle diffusing in a one-dimensional random potential, *Phys. Rev. E*, 73:051102, 2006.
[34] J. Sirignano and K. Spiliopoulos, DGM: A deep learning algorithm for solving partial differential equations, *J. Comput. Phys.*, 375:1339–1364, 2018.
[35] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, Attention is all you need, *arXiv:1706.03762*, 2017.
[36] B. Wang, S. M. Anthony, S. C. Bae, and S. Granick, Anomalous yet Brownian, *Proc. Natl. Acad. Sci. USA*, 106(36):15160–15164, 2009.

[37] B. Wang, J. Kuo, S. C. Bae, and S. Granick, When Brownian diffusion is not Gaussian, *Nat. Mater*, 11:481–485, 2012.
[38] X. D. Wang, Y. Chen, and W. H. Deng, Feynman-Kac equation revisited, *Phys. Rev. E*, 98:052114, 2018.
[39] E. R. Weeks, J. C. Crocker, A. C. Levitt, A. Schofield, and D. A. Weitz, Three-dimensional direct imaging of structural relaxation near the colloidal glass transition, *Science*, 287:627–631, 2000.
[40] Y. Zang, G. Bao, X. Ye, and H. Zhou, Weak adversarial networks for high-dimensional partial differential equations, *J. Comput. Phys.*, 411:109409, 2020.
[41] T. Zhou, H. Wang, and W. H. Deng, Feynman-Kac equation for Brownian non-Gaussian polymer diffusion, *J. Phys. A: Math. Theor.*, 2024. DOI 10.1088/1751-8121/ad57b4