SIAM J. SCI. COMPUT.
Vol. 48, No. 5, pp. C1380–C1402
© 2026 Society for Industrial and Applied Mathematics

Downloaded 10/27/26 to 161.81.191.158 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

DEEP LEARNING FOR HIGH-DIMENSIONAL PDES ON MANIFOLD WITH NEUMANN AND DIRICHLET BOUNDARY CONDITIONS*

HENG WANG† AND WEIHUA DENG‡

Abstract. The deep backward stochastic differential equation (BSDE) method is a deep learning algorithm for solving high-dimensional partial differential equations (PDEs) in the whole space, which is trained using the sampled trajectories of microscopic stochastic processes; however, generally it is incapable of solving the problems on a manifold or with boundary constraints. To overcome this issue, we propose a strategy compliant with BSDE theory in this paper, termed the Constraint-Informed BSDE method. The key idea of the Constraint-Informed BSDE method is to first construct appropriate stochastic processes based on the physical implications of boundary conditions and manifold constraints, and further derive the BSDEs that the solutions to the corresponding PDEs satisfy. Specifically, we derive the BSDEs satisfied by the solutions of PDEs with Neumann and Dirichlet boundary conditions on multifold sphere and in high-dimensional domain. Finally, we demonstrate the performance of the Constraint-Informed BSDE method through extensive numerical experiments. Since there is no need to explicitly incorporate boundary conditions into the loss function, Constraint-Informed BSDE method exhibits sufficient accuracy, stability, and strong robustness.

Key words. deep BSDE method, boundary conditions, manifold constraint

MSC codes. 60J76, 60A10, 35R11, 60H15, 68T07

DOI. 10.1137/25M1826822



See reproducibility of computational results at end of the article.

1. Introduction.

Partial differential equations (PDEs) are important tools for modeling real-world phenomena, having a wide range of applications in various fields [17], such as bioscience, fluid mechanics, meteorology, and finance. We need to develop appropriate algorithms to effectively solve the built PDEs. The traditional numerical methods, such as finite difference methods, finite element methods, and spectral methods, provide effective and highly accurate approximations for the solutions of PDEs; however, their computational complexity grows exponentially with the increase of dimensions. Benefiting from the powerful high-dimensional representation capability of deep neural networks, a large number of deep learning solvers have been developed. In these algorithms, neural networks are typically used as approximation functions of high-dimensional solutions and are trained by constructing loss functions based on PDEs. Inspired by traditional numerical schemes, there are four categories of popular deep learning frameworks. The first framework is the physics-informed neural network

*Submitted to the journal's Machine Learning Methods for Scientific Computing section December 1, 2025; accepted for publication (in revised form) June 8, 2026; published electronically October 27, 2026.

https://doi.org/10.1137/25M1826822

Funding: This work was supported by the National Natural Science Foundation of China under grant 12225107, the Major Science and Technology Projects in Gansu Province-Leading Talents in Science and Technology under grant 23ZDKA0005, the Innovative Groups of Basic Research in Gansu Province under grant 22JR5RA391, the Tianyuan Fund for Mathematics of the National Natural Science Foundation of China 12426105, and Lanzhou Talent Work Special Fund.

†School of Mathematics and Statistics, State Key Laboratory of Natural Product Chemistry, Lanzhou University, Lanzhou 730000, China (wheng2025@lzu.edu.cn)

‡Corresponding author. School of Mathematics and Statistics, State Key Laboratory of Natural Product Chemistry, Lanzhou University, Lanzhou 730000, China (dengwh@lzu.edu.cn).

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 10/27/26 to 161.81.191.158 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

(PINN) [15], similar to the least square method, which constructs the loss function using the L_2 norm of the residual between the two sides of the PDE. The second is the deep Ritz method [3], obtaining the loss function based on the energy functional of the PDE. The third one targets PDEs that do not have an energy functional, solving the equation through adversarial learning [19], which is similar to the Petrov–Galerkin method. In this paper, we focus on the fourth framework, termed the deep backward stochastic differential equation (BSDE) method [6], which embodies a strong machine learning essence: first, it requires deriving the BSDE that the solution to the PDE satisfies by constructing an appropriate microscopic stochastic process; second, it uses sampled trajectories of the stochastic process to train the BSDE. Subsequently, Wang and Deng extend the deep BSDE method to solve macroscopic equations for polymer center of mass [16] and develop deep learning algorithms for fractional PDEs based on BSDE theory for jump process [1].

The deep BSDE method focuses on a class of high-dimensional semilinear parabolic PDEs

$$\begin{aligned}\frac{\partial}{\partial t}u(t,x)+\nabla_x u(t,x)\cdot\mu(t,x)+\frac{1}{2}\mathrm{Tr}\left(\sigma\sigma^T(t,x)\left(\mathrm{Hess}_x u\right)(t,x)\right)\\ +f\left(t,x,u(t,x),(\sigma^T\nabla_x u)(t,x)\right)=0\end{aligned}\tag{1.1}$$

with terminal condition $u(T,x)=u_0(x)$. The key idea of deep BSDE method is to transform (1.1) into the forward-backward stochastic differential equations (FBSDEs)

$$\begin{cases}X(t)=X(0)+\displaystyle\int_0^t\mu(s,X(s))ds+\int_0^t\sigma(s,X(s))dB(s),\\ Y(t)=Y(T)+\displaystyle\int_t^T f\left(s,X(s),Y(s),Z(s)\right)ds-\int_t^T[Z(s)]^T dB(s),\end{cases}\tag{1.2}$$

which share the same solutions $(Y(t),Z(t))=(u(t,X(t)),(\sigma^T\nabla_x u)(t,X(t)))$ of (1.1) and can be solved using deep learning. Benefiting from the theory of existence and uniqueness [11] of BSDE (1.2), one can solve for the solution at any target position of interest without obtaining the global solution; furthermore, there is no need for differential operations (because of the existence and uniqueness of $Z(t)=(\sigma^T\nabla_x u)(t,X(t))$), which can be directly approximated by a neural network and significantly improves algorithm efficiency and reduces computational costs. As for the convergence of the deep BSDE method, one can refer to [7, 4]. Benefiting from the relationship between the forward stochastic process $X(t)$ and the operators in the PDE, this method can be combined with observation data to enhance the model's extensibility [1]. However, when $u(t,x)$ is the solution of a problem with boundary constraints or on a manifold, this method fails because the stochastic process $X(t)$ moves in the whole space.

To effectively solve PDEs with boundary constraints on a manifold, we propose the Constraint-Informed BSDE method, which is consistent with the theory of BSDEs with stopping time [14, 11, 2, 13, 12]. The key idea of the Constraint-Informed BSDE method is to construct appropriate stochastic processes based on the physical implications of boundary conditions and manifold constraints, and further derive the BSDEs that the solutions to the corresponding PDEs satisfy. Specifically, we derive the BSDEs satisfied by the solutions of PDEs with Neumann and Dirichlet boundary conditions on multifold sphere and in high-dimensional domain. Finally, we demonstrate the performance of the Constraint-Informed BSDE method through extensive numerical experiments.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 10/27/26 to 161.81.191.158 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

The remainder of the paper is organized as follows. In section 2, we will elaborate on the main content of the Constraint-Informed BSDE method, which specifically includes the derivation of BSDEs for PDEs with Neumann and Dirichlet boundary conditions on mulitfold sphere and in high-dimensional domain, and design the associated deep learning algorithms. Subsequently, in section 3, we demonstrate the performance of the proposed algorithms through extensive numerical experiments. Finally, the paper is concluded in section 4.

2. Methodology. Let $x(\theta,\phi) = [\sin\theta\cos\phi, \sin\theta\sin\phi, \cos\theta]^T$ be the coordinate transformation that maps the polar coordinates (θ,ϕ) on the unit sphere to Cartesian coordinates. We consider the parabolic PDE

$$\begin{aligned}(2.1)\quad &\frac{\partial}{\partial t}p(t,x_S,x_E) + \nabla_E p(t,x_S,x_E)\cdot\mu(t,x_E) + \frac{1}{2}\mathrm{Tr}\left(\sigma\sigma^T(t,x_E)\left(\mathrm{Hess}_E p\right)(t,x_S,x_E)\right)\\ &\quad + \Delta_S p(t,x_S,x_E) + f\left(t,x_S,x_E,p(t,x_S,x_E),\left(\sigma^T\nabla_E p\right)(t,x_S,x_E)\right) = 0\end{aligned}$$

with the Neumann boundary condition $\frac{\partial}{\partial n}p(t,x_S,x_E)|_{(x_S,x_E)\in\Gamma_1} = h(t,x_S,x_E)$, the Dirichlet boundary condition $p(t,x_S,x_E)|_{(x_S,x_E)\in\Gamma_2} = g(t,x_S,x_E)$, and the terminal condition $p(T,x_S,x_E) = g(T,x_S,x_E)$. Here, $t\in[0,T]$ represents the time variable, $x_S = (x(\theta_1,\phi_1),\dots,x(\theta_{d_S},\phi_{d_S}))\in\Omega_{S_1}\times\cdots\times\Omega_{S_{d_S}}\subset\partial B(0,1)^{d_S}$ denotes the d_S-fold sphere variable, $x_E\in\Omega_E\subset\mathbb{R}^{d_E}$ is the d_E-dimensional space variable, Γ_1 and Γ_2 are the boundaries of $\Omega = \Omega_{S_1}\times\cdots\times\Omega_{S_{d_S}}\times\Omega_E$, μ and σ are known d_E-dimensional vector-valued and $d_E\times d_E$-dimensional matrix-valued functions, f is a known nonlinear function, $\nabla_E p$ and $\mathrm{Hess}_E p$ denote the gradient and the Hessian of function p with respect to x_E, Tr denotes the trace of a matrix, and Δ_S is defined as the summation of the Laplace–Beltrami operators on spheres

$$\Delta_{S_i}u(\theta_i,\phi_i) = \frac{1}{\sin\theta_i}\frac{\partial}{\partial\theta_i}\left(\sin\theta_i\frac{\partial}{\partial\theta_i}u(\theta_i,\phi_i)\right) + \frac{1}{\sin^2\theta_i}\frac{\partial^2}{\partial\phi_i^2}u(\theta_i,\phi_i)$$

with diffusion coefficient D_i, which takes the form

$$\Delta_S u(x(\theta_1,\phi_1),\dots,x(\theta_{d_S},\phi_{d_S})) = \sum_{i=1}^{d_S} D_i\Delta_{S_i}u(x(\theta_1,\phi_1),\dots,x(\theta_{d_x},\phi_{d_x})).$$

In this section, we will elaborate on the details of the Constraint-Informed BSDE method, which is a BSDE-based deep learning algorithm for solving (2.1). To provide readers with an overview of the method as a whole, we first present the overall architecture of the Constraint-Informed BSDE method in Figure 1.

2.1. Constraint-Informed FBSDEs. We focus on the FBSDEs satisfied by the solution to (2.1), which constitutes the first step in designing the deep learning algorithm and requires overcoming two issues: the first is the singularity issue of $1/\sin\theta$ (as $\theta\to 0,\pi$) in the Laplace–Beltrami operator; the second is that (2.1) is defined on the bounded domain, which prevents us from directly conducting the derivation merely by defining a forward process of the form given in (1.2). In this part, we first outline our key ideas for overcoming the singularity problem and the boundedness problem, and then, we derive the FBSDEs; finally, we propose a reformulation of the obtained FBSDEs to theoretically ensure its correctness.

Singularity. The singularity problem of the Laplace–Beltrami operator at $x(\theta\in\{0,\pi\},\phi)$ actually depends on how the polar coordinate system is constructed and is

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

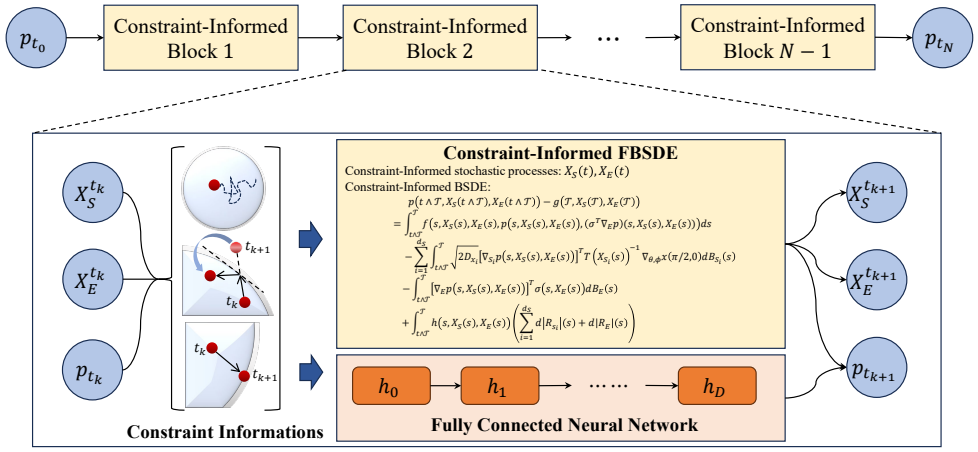


FIG. 1. *The overall architecture of Constraint-Informed BSDE method. Starting from* t_0, p_{t_0}, *for each other time layer, the Constraint-Informed BSDE block receives the* $p_{t_k} = p(t_k, X_S^{t_k}, X_E^{t_k})$ *(output from the previous block),* $X_S^{t_k} = X_S(t_k)$, *and* $X_E^{t_k} = X_E(t_k)$. *Based on the iterative scheme constructed by the Constraint-Informed FBSDEs, these modules update forward by one step in conjunction with a fully connected neural network. Finally, the neural network outputs* $p_{t_N} = \hat{p}(\mathcal{T}, X_S(\mathcal{T}), X_E(\mathcal{T}); \theta)$; *the discrepancy between* p_{t_N} *and the true value is used to construct the loss function, which in turn optimizes the network parameters.*

not a "physical" problem. The points farthest from the singular points are $(\theta, \phi) \in \{\pi/2\} \times [0, 2\pi)$, so the key idea of addressing the singularity issue is to parameterize the target point at the polar angle $\theta = \pi/2$ with a default azimuthal angle $\phi = 0$. To "incorporate" this spherical information into the FBSDEs, we define a rotation matrix $T(x(\theta, \phi)) = R_y(\theta - \pi/2)R_z(\phi)$, where $R_z(\phi)$ and $R_y(\theta)$ denote the rotation matrices that rotate clockwise by ϕ degrees around the z-axis and by θ degrees around the y-axis, respectively, with the form

$$R_z(\phi) = \begin{bmatrix} \cos\phi & \sin\phi & 0 \\ -\sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad R_y(\theta) = \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix}. \tag{2.2}$$

The matrix $T(x(\theta, \phi))$ is to rotate $x(\theta, \phi)$ to the fixed point $x(\pi/2, 0)$. It is noted that for some given rotation matrices $T(\hat{x}_i)$, $i = 1, \ldots, d_S$, if we denote

$$\hat{x}(x_S) = (T(\hat{x}_1)^{-1} x(\theta_1, \phi_1), \ldots, T(\hat{x}_{d_S})^{-1} x(\theta_{d_S}, \phi_{d_S}))$$

and $p_{\hat{x}}(t, x_S, x_E) = p(t, \hat{x}(x_S), x_E)$, then $p_{\hat{x}}$ satisfies

$$\begin{aligned} &\frac{\partial}{\partial t} p_{\hat{x}}(t, x_S, x_E) + \nabla_E p_{\hat{x}}(t, x_S, x_E) \cdot \mu(t, x_E) + \frac{1}{2} \mathrm{Tr}\left(\sigma\sigma^T(t, x_E) \left(\mathrm{Hess}_E p_{\hat{x}}\right)(t, x_S, x_E)\right) \\ &\quad + \Delta_S p_{\hat{x}}(t, x_S, x_E) + f\left(t, \hat{x}(x_S), x_E, p_{\hat{x}}(t, x_S, x_E), \left(\sigma^T \nabla_E p_{\hat{x}}\right)(t, x_S, x_E)\right) = 0 \end{aligned} \tag{2.3}$$

with the Neumann boundary condition $\frac{\partial}{\partial n} p_{\hat{x}}(t, x_S, x_E)|_{(\hat{x}(x_S), x_E) \in \Gamma_1} = h(t, \hat{x}(x_S), x_E)$, the Dirichlet boundary condition $p_{\hat{x}}(t, x_S, x_E)|_{(\hat{x}(x_S), x_E) \in \Gamma_2} = g(t, \hat{x}(x_S), x_E)$, and the terminal condition $p_{\hat{x}}(T, x_S, x_E) = g(T, \hat{x}(x_S), x_E)$. Property (2.3) is crucial for overcoming the singularity, as the singularity of this equation at $x(\theta_i, \phi_i) = x(\pi/2, 0)$,

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 10/27/26 to 161.81.191.158 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

Downloaded 10/27/26 to 161.81.191.158 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

$i = 1, \ldots, d_S$, disappears, making it a key step in the subsequent derivation of the BSDE.

Boundedness. There are two constraint conditions that restrict the solution of (2.1) to a bounded domain: the Neumann boundary condition and the Dirichlet boundary condition. The key idea to address the constraint imposed by the Neumann boundary condition is to extend (2.1) to the whole space by leveraging its physical interpretation as a prescribed net flux across the boundary. Therefore, for the physical domain Ω, we may assume that there exist some mirrored systems in the mirror-symmetric domains Ω_{x_b} with respect to the tangent planes (which have unit inward normal vectors $n(x_b)$) at all boundary points $x_b \in \Gamma_1$. This results in a mirror-connected system of the original region Ω with respect to the boundary point x_b, which can be described by the mirror reflection map $\mathcal{F}(x) = x + 2((x_b - x) \cdot n(x_b))n(x_b)$ for $x \in \Omega_{x_b}$. The Dirichlet boundary condition means that the value of p at the boundary point $(x_S, x_E) \in \Gamma_2$ is enforced to be $g(t, x_S, x_E)$, and the location of reaching Γ_2 can be recorded by the first hitting time of the forward process.

Now, we can begin to define the Constraint-Informed stochastic processes, $X_S(t)$ and $X_E(t)$, corresponding to the variables x_S and x_E, respectively. Among them, $X_S(t) = [X_{S_1}(t), \ldots, X_{S_{d_S}}(t)]^T$ is a stochastic process on a d_S-fold spherical surface, driven by a Brownian motion $B_S(t) = [B_{S_1}(t), \ldots, B_{S_{d_S}}(t)]^T$ with components $B_{S_i}(t) = [B_{\theta_i}(t), B_{\phi_i}(t)]^T$. For all $i = 1, \ldots, d_S$, the component process $X_{S_i}(t)$ satisfies $X_{S_i}(0) = \xi_{S_i} \in \Omega_{S_i}$,

$$\tilde{X}_{S_i}(t + dt) = X_{S_i}(t) + T(X_{S_i}(t))^{-1} dB_{S_i}(\pi/2, 0)(t), \tag{2.4}$$

and

$$X_{S_i}(t + dt) = \tilde{X}_{S_i}(t + dt) + n_{S_i}(x^b_{S_i}(t)) d|R_{S_i}|(t), \tag{2.5}$$

where $dB_{S_i}(\pi/2, 0)(t) = x(\pi/2 + \sqrt{2D_i} dB_{\theta_i}(t), \sqrt{2D_i} dB_{\phi_i}(t)) - x(\pi/2, 0)$, $d|R_{S_i}|(t) = \mathbb{I}_{\tilde{X}_{S_i}(t+dt) \notin \overline{\Omega}_{S_i}} 2(x^b_{S_i}(t) - \tilde{X}_{S_i}(t + dt)) \cdot n_{S_i}(X_{S_i}(t))$, $x^b_{S_i}(t)$ denotes the boundary point passed when moving from $X_{S_i}(t)$ to $\tilde{X}_{S_i}(t+dt)$, and $n_{S_i}(x_{S_i})$ is the unit inward normal vector of domain Ω_{S_i} at the boundary point $x_{S_i} \in \partial\Omega_{S_i}$. The stochastic process $X_E(t)$ is driven by a d_E-dimensional Brownian motion $B_E(t)$ and satisfies $X_E(0) = \xi_E \in \Omega_E$,

$$\tilde{X}_E(t + dt) = X_E(t) + \mu(t, X_E(t))dt + \sigma(t, X_E(t))dB_E(t), \tag{2.6}$$

and

$$X_E(t + dt) = \tilde{X}_E(t + dt) + n_E(x^b_E(t)) d|R_E|(t), \tag{2.7}$$

where $d|R_E|(t) = \mathbb{I}_{\tilde{X}_E(t+dt) \notin \overline{\Omega}_E} 2(x^b_E(t) - \tilde{X}_E(t + dt)) \cdot n_E(X_E(t))$, $x^b_E(t)$ denotes the boundary point passed when moving from $X_E(t)$ to $\tilde{X}_E(t + dt)$, and $n_E(x_E)$ is the unit inward normal vector of domain Ω_E at the boundary point $x_E \in \partial\Omega_E$.

Here, (2.4) and (2.6) indicate that $X_S(t)$ and $X_E(t)$ undergo stochastic motion driven by the random driving terms. In (2.4), the rotation matrix $T(X_{S_i}(t))^{-1}$ preceding the driving term, combined with property (2.3), can overcome the singularity of the Laplace–Beltrami operators in the subsequent derivation of the BSDE. Therefore, (2.5) and (2.7) imply that if the positions $X_{S_i}(t)$ and $X_E(t)$ after one step of random motion lie outside the closure of the corresponding domains $\overline{\Omega}_{S_i}$ and $\overline{\Omega}_E$, the mirror reflections are applied to them, and then, the new positions are taken as the postmotion positions of $X_{S_i}(t + dt)$ and $X_E(t + dt)$; otherwise, $\tilde{X}_{S_i}(t + dt)$ and $\tilde{X}_E(t+dt)$ are directly used as their own postmotion positions. It should be noted that

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 10/27/26 to 161.81.191.158 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

the stochastic processes driven by Brownian motion are continuous; therefore, exiting the regions $\tilde{X}_{S_i}(t+dt) \notin \overline{\Omega}_{S_i}$ and $\tilde{X}_E(t+dt) \notin \overline{\Omega}_E$ implies that at the previous time step, the processes were on the boundary of the regions, i.e., $x^b_{S_i}(t) = X_{S_i}(t) \in \partial\Omega_{S_i}$ and $x^b_E(t) = X_E(t) \in \partial\Omega_E$.

Remark 2.1. In fact, if we substitute (2.4) and (2.6) into (2.5) and (2.7), respectively, and then integrate with respect to time t, we can obtain

$$
\begin{cases}
X_{S_i}(t) = \xi_{S_i} + \displaystyle\int_0^t T(X_{S_i}(s))^{-1} dB_{S_i}(\pi/2, 0)(s) + R_{S_i}(t), \\
X_E(t) = \xi_E + \displaystyle\int_0^t \mu(s, X_E(s))ds + \int_0^t \sigma(s, X_E(s))dB_E(s) + R_E(t),
\end{cases} \tag{2.8}
$$

where $R_{S_i}(t) = \int_0^t n_{S_i}(X_{S_i}(s))d|R_{S_i}|(s)$, and $R_E(t) = \int_0^t n_E(X_E(s))d|R_E|(s)$. Equation (2.8) is the SDE with reflection. Under the assumption that the boundary $\partial\Omega$ is of class C^2, and that there exist $C_\mu \in \mathbb{R}$ and $C_\sigma > 0$ such that for all $u, v \in \mathbb{R}^{d_E}$,

$$
\begin{aligned}
&\text{(i)} \quad \langle u - v, \mu(t,u) - \mu(t,v)\rangle \le C_\mu |u - v|^2, \\
&\text{(ii)} \quad |\sigma(t,u) - \sigma(t,v)| \le C_\sigma |u - v|,
\end{aligned}
$$

one can obtain that (2.8) has unique solution pairs $(X_{S_i}(t), R_{S_i}(t))$ and $(X_E(t), R_E(t))$ (see, e.g., [12]). Therefore, (2.4)–(2.7) provide an intuitive representation of the reflected SDE, making it easier to understand and facilitating the design of algorithms.

To formalize the stopping event of $(X_S(t), X_E(t))$ upon reaching the boundary Γ_2, we define the first hitting time $T_f = \inf\{s; (X_S(t), X_E(t)) \in \Gamma_2\}$ and denote $\mathcal{T} = T \wedge T_f$. Driven by the Constraint-Informed stochastic processes $X_S(t)$ and $X_E(t)$, one can now derive the corresponding Constraint-Informed BSDE (see Appendix A)

$$
\begin{aligned}
&p(t\wedge\mathcal{T}, X_S(t\wedge\mathcal{T}), X_E(t\wedge\mathcal{T})) - g(\mathcal{T}, X_S(\mathcal{T}), X_E(\mathcal{T})) \\
&\quad = \int_{t\wedge\mathcal{T}}^{\mathcal{T}} f\left(s, X_S(s), X_E(s), p(s, X_S(s), X_E(s)), (\sigma^T \nabla_E p)(s, X_S(s), X_E(s))\right) ds \\
&\qquad - \sum_{i=1}^{d_S} \int_{t\wedge\mathcal{T}}^{\mathcal{T}} \sqrt{2D_i} [\nabla_{S_i} p(s, X_S(s), X_E(s))]^T T(X_{S_i}(s))^{-1} \nabla_{\theta,\phi} x(\pi/2, 0) dB_{S_i}(s), \\
&\qquad - \int_{t\wedge\mathcal{T}}^{\mathcal{T}} [\nabla_E p(s, X_S(s), X_E(s))]^T \sigma(s, X_E(s)) dB_E(s) \\
&\qquad + \int_{t\wedge\mathcal{T}}^{\mathcal{T}} h(s, X_S(s), X_E(s)) \left(\sum_{i=1}^{d_S} d|R_{S_i}|(s) + d|R_E|(s) \right),
\end{aligned} \tag{2.9}
$$

where $\nabla_{S_i} p$ denotes the gradient of function p with respect to $x(\theta_i, \phi_i)$, and $\nabla_{\theta,\phi} x$ is for the gradient of function $x(\theta, \phi)$ with respect to (θ, ϕ). It should be pointed out that when $f(t, x_S, x_E, p, \nabla_E p) = c(t, x_S, x_E)p$, the solution $p(t, x_S, x_E)$ has the stochastic representation

$$
\begin{aligned}
p(t, x_S, x_E) = \mathbb{E}\Bigg[& e^{\int_t^{\mathcal{T}} c(s, X_S^{x_S,t}(s), X_E^{x_E,t}(s))ds} g(\mathcal{T}, X_S^{x_S,t}(\mathcal{T}), X_E^{x_E,t}(\mathcal{T})) \\
& + \int_t^{\mathcal{T}} e^{\int_t^s c(r, X_S^{x_S,t}(r), X_E^{x_E,t}(r))dr} h(s, X_S^{x_S,t}(s), X_E^{x_E,t}(s)) d|R|(s) \Bigg],
\end{aligned} \tag{2.10}
$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 10/27/26 to 161.81.191.158 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

where $X_S^{x_S,t}(s)$ is the solution of (2.4) and (2.5) with $X_S(t) = x_S$, $X_E^{x_E,t}(s)$ is solution of (2.6) and (2.7) with $X_E(t) = x_E$, and $d|R|(s) = \sum_{i=1}^{d_S} d|R_{S_i}|(s) + d|R_E|(s)$.

Remark 2.2. Equation (2.9) can lead to a BSDE with stopping time, which has the form

$$
\begin{aligned}
Y(t\wedge\mathcal{T}) = {} & g(\mathcal{T}, X_S(\mathcal{T}), X_E(\mathcal{T})) + \int_{t\wedge\mathcal{T}}^{\mathcal{T}} f(s, X_S(s), X_E(s), Y(s), Z_E(s))ds \\
& - \sum_{i=1}^{d_S} \int_{t\wedge\mathcal{T}}^{\mathcal{T}} [Z_{S_i}(s)]^T dB_{S_i}(s) - \int_{t\wedge\mathcal{T}}^{\mathcal{T}} [Z_E(s)]^T dB_E(s) \\
& + \int_{t\wedge\mathcal{T}}^{\mathcal{T}} h(s, X_S(s), X_E(s)) \left(\sum_{i=1}^{d_S} d|R_{S_i}|(s) + d|R_E|(s) \right).
\end{aligned} \tag{2.11}
$$

The well-posedness of BSDE with stopping time and its connection with parabolic PDEs with Neumann and Dirichlet boundary conditions have been studied (cf., e.g., [12, 2, 14, 13]). If $p \in C^{1,2}([0,T]\times\overline{\Omega})$ is a classical solution of (2.1), then the derivation of (2.9) can be carried out directly using Itô's formula. This forward reasoning shows that the solution of (2.1) belongs to the set of solutions of the BSDE (2.11). Conversely, the existence and uniqueness for BSDE (2.11) implies that the solution obtained by directly solving this BSDE coincides with that of (2.1).

2.2. Constraint-Informed BSDE method. In this subsection, we proceed to design the Constraint-Informed BSDE method based on the obtained FBSDEs (2.4)–(2.7) and (2.9). For a given partition of the time interval $[0,T] : 0 = t_0 < t_1 < \cdots < t_N = T$, one can apply temporal discretizations to (2.4)–(2.7), which are

$$
\tilde{X}_{S_i}(t_{k+1}) \approx X_{S_i}(t_k) + T(X_{S_i}(t_k))^{-1} \Delta B_{S_i}\,(\pi/2, 0)\,(t_k), \tag{2.12}
$$

$$
X_{S_i}(t_{k+1}) \approx \tilde{X}_{S_i}(t_{k+1}) + n_{S_i}(x^b_{S_i}(t_k))\Delta|R_{S_i}|(t_k), \tag{2.13}
$$

$$
\tilde{X}_E(t_{k+1}) \approx X_E(t_k) + \mu(t_k, X_E(t_k))\Delta t_k + \sigma(t_k, X_E(t_k))\Delta B_E(t_k), \tag{2.14}
$$

and

$$
X_E(t_{k+1}) \approx \tilde{X}_E(t_{k+1}) + n_E(x^b_E(t_k))\Delta|R_E|(t_k). \tag{2.15}
$$

By determining whether the line connecting the positions at two time steps passes through the boundary Γ_2, we can obtain an approximation of the first hitting time

$$
\begin{aligned}
T_f \approx \min \Big\{ t_k; \{ & s[\tilde{X}_S(t_{k+1}), \tilde{X}_E(t_{k+1})]^T \\
& + (1-s)[X_S(t_k), X_E(t_k)]^T; s \in [0,1] \} \cap \Gamma_2 \neq \emptyset \Big\} \cup \{+\infty\}.
\end{aligned}
$$

Then, we can apply a temporal discretization to (2.9) to obtain

(2.16)

$$
\begin{aligned}
& p(t_{k+1}, X_S(t_{k+1}), X_E(t_{k+1})) - p(t_k, X_S(t_k), X_E(t_k)) \\
& \approx \Big[- f\left(t_k, X_S(t_k), X_E(t_k), p(t_k, X_S(t_k), X_E(t_k)), \left(\sigma^T \nabla_E\right) p(t_k, X_S(t_k), X_E(t_k))\right) \Delta t_k
\end{aligned}
$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

$$+\sum_{i=1}^{d_S}\sqrt{2D_i}[\nabla_{S_i}p(t_k,X_S(t_k),X_E(t_k))]^T T(X_{S_i}(t_k))^{-1}\nabla_{\theta,\phi}x(\pi/2,0)\Delta B_{S_i}(t_k)$$
$$+[\nabla_E p(t_k,X_S(t_k),X_E(t_k))]^T\sigma(t_k,X_E(t_k))\Delta B_E(t_k)$$
$$-h\left(t_k,X_S(t_k),X_E(t_k)\right)\left(\sum_{i=1}^{d_S}\Delta|R_{S_i}|(t_k)+\Delta|R_E|(t_k)\right)\Bigg]\mathbb{I}_{\{t_k\le T_f\}}.$$

In the above discrete schemes, $\Delta t_k = t_{k+1}-t_k$, $\Delta B_E(t_k) = B_E(t_{k+1})-B_E(t_k)$, $\Delta B_{S_i}(t_k) = [\Delta B_{\theta_i}(t_k),\Delta B_{\phi_i}(t_k)]^T$, $\Delta B_{\theta_i}(t_k) = B_{\theta_i}(t_{k+1})-B_{\theta_i}(t_k)$, $\Delta B_{\phi_i}(t_k) = B_{\phi_i}(t_{k+1})-B_{\phi_i}(t_k)$,

$$\Delta B_{S_i}(\pi/2,0)(t_k) = x(\pi/2+\sqrt{2D_i}\Delta B_{\theta_i}(t_k),\sqrt{2D_i}\Delta B_{\phi_i}(t_k)) - x(\pi/2,0), \tag{2.17}$$

$$\Delta|R_{S_i}|(t_k) = \mathbb{I}_{\tilde{X}_{S_i}(t_{k+1})\notin\overline{\Omega}_{S_i}}2(x^b_{S_i}(t_k)-\tilde{X}_{S_i}(t_{k+1}))\cdot n_{S_i}(x^b_{S_i}(t_k)), \tag{2.18}$$

$$\Delta|R_E|(t_k) = \mathbb{I}_{\tilde{X}_E(t_{k+1})\notin\overline{\Omega}_E}2(x^b_E(t_k)-\tilde{X}_E(t_{k+1}))\cdot n_E(x^b_E(t_k)), \tag{2.19}$$

and $x^b_{S_i}(t_k)$ and $x^b_E(t_k)$ denote the boundary points passed when moving from $X_{S_i}(t_k)$ to $\tilde{X}_{S_i}(t_{k+1})$ and from $X_E(t_k)$ to $\tilde{X}_E(t_{k+1})$, respectively, which can also be approximated as the boundary points closest to $\tilde{X}_{S_i}(t_{k+1})$ and $\tilde{X}_E(t_{k+1})$. The trajectories of $\{X_{S_1}(t_k),\dots,X_{S_{d_S}}(t_k),X_E(t_k)\}_{0\le k\le N}$ can be sampled using (2.12)–(2.15), and the functions $(x_S,x_E)\mapsto p(0,x_S,x_E)$ and $(x_S,x_E)\mapsto\nabla_{x_S,x_E}p(t_k,x_S,x_E)$ can be approximated using two different neural networks

$$p(0,x_S,x_E)\approx\psi_p(x_S,x_E;\theta_p)$$

and

$$\nabla_{x_S,x_E}p(t_k,x_S,x_E)\approx\psi_{\nabla p}(x_S,x_E;\theta_{\nabla p_k}),$$

for $k=0,\dots,N-1$, where θ_p and $\{\theta_{\nabla p_k}\}_{0\le k\le N-1}$ are corresponding trainable parameters. The terminal value $g(\mathcal{T},X_S(\mathcal{T}),X_E(\mathcal{T}))$ can be approximated using (2.16), denoted by $\hat{p}(\mathcal{T},X_S(\mathcal{T}),X_E(\mathcal{T});\theta)$ with parameters $\theta=\{\theta_p,\theta_{\nabla p_0},\dots,\theta_{\nabla p_{N-1}}\}$. Finally, we can train the neural network by minimizing the loss function

$$\text{Loss}(\theta) = \mathbb{E}\left[|\hat{p}(\mathcal{T},X_S(\mathcal{T}),X_E(\mathcal{T});\theta)-g(\mathcal{T},X_S(\mathcal{T}),X_E(\mathcal{T}))|^2\right]. \tag{2.20}$$

The above discussion yields the Constraint-Informed BSDE method, as detailed in Algorithm 2.1. The full architecture of Constraint-Informed BSDE method is shown in Figure 1.

Remark 2.3. It should be noted that (2.12)–(2.20) can be regarded as a Monte Carlo approximation of variational problem

$$\inf_{Y(0),\{Z_S(t),Z_E(t)\}_{0\le t\le T}}\mathbb{E}\left[|g(\mathcal{T},X_S(\mathcal{T}),X_E(\mathcal{T}))-Y(T)|^2\right] \tag{2.21}$$
$$\text{s.t. } X_{S_i}(t)=\xi_{S_i}+\int_0^t T(X_{S_i}(s))^{-1}dB_{S_i}(\pi/2,0)(s)+R_{S_i}(t),\ i=1,\dots,d_S,$$
$$X_E(t)=\xi_E+\int_0^t\mu(s,X_E(s))ds+\int_0^t\sigma(s,X_E(s))dB_E(s)+R_E(t),$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 10/27/26 to 161.81.191.158 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

Downloaded 10/27/26 to 161.81.191.158 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

Algorithm 2.1. The algorithm of the Constraint-Informed BSDE method.

Input: Neural networks $\psi_p(x_S, x_E|\theta_p)$ and $\{\psi_{\nabla p}(x_S, x_E|\theta_{\nabla p_k})\}_{0\le k\le N-1}$, maximum number of iterations $\mathcal{N}$, the partition of the time interval $[t,T]: t=t_0<t_1<\cdots<t_{N-1}<t_N=T$, and learning rate α

Output: Neural network $\psi_p(x_S, x_E|\theta_p)$

Initialize step $=1$
while step $\le \mathcal{N}$ **do**
  Generate $X_S(t_0)=\xi_S,\ X_E(t_0)=\xi_E$
  $p(t_0, X_S(t_0), X_E(t_0)) \leftarrow \psi(X_S(t_0), X_E(t_0)|\theta_p)$
  for k from 0 to $N-1$ **do**
    Generate $\Delta B_{S_1}(t_k), \ldots, \Delta B_{S_{d_S}}(t_k), \Delta B_E(t_k)$
    $\Delta B_{S_i}(\pi/2,0)(t_k) \leftarrow x(\pi/2+\sqrt{2D_{x_i}}\Delta B_{\theta_i}(t_k), \sqrt{2D_{x_i}}\Delta B_{\phi_i}(t_k)) - x(\pi/2,0)$
    $\nabla_{x_S,x_E} p(t_k, X_S(t_k), X_E(t_k)) \leftarrow \psi_{\nabla p}(X_S(t_k), X_E(t_k)|\theta_{\nabla p_k})$
    $\{\tilde{X}_{S_i}(t_{k+1})\}_{1\le i\le d_S} \xleftarrow{(2.12)} \{X_{S_i}(t_k), \Delta B_{S_i}(\pi/2,0)(t_k)\}_{1\le i\le d_S}$
    $\{\Delta|R_{S_i}|(t_k)\}_{1\le i\le d_S} \xleftarrow{(2.18)} \{\tilde{X}_{S_i}(t_{k+1}), x^b_{S_i}(t_k), n_{S_i}(x^b_{S_i}(t_k))\}_{1\le i\le d_S}$
    $\tilde{X}_E(t_{k+1}) \xleftarrow{(2.14)} X_E(t_k), \Delta B_E(t_k)$
    $\Delta|R_E|(t_k) \xleftarrow{(2.19)} \tilde{X}_E(t_{k+1}), x^b_E(t_k), n_E(x^b_E(t_k))$
    $p(t_{k+1}, X_S(t_{k+1}), X_E(t_{k+1})) \xleftarrow{(2.16)} p(t_k, X_S(t_k), X_E(t_k)),$
      $\nabla p(t_k, X_S(t_k), X_E(t_k)), X_S(t_k), X_E(t_k),$
      $\{\Delta B_{S_i}(t_k)\}_{1\le i\le d_S}, \Delta B_E(t_k),$
      $\{\Delta|R_{S_i}|(t_k)\}_{1\le i\le d_S}, \Delta|R_E|(t_k)$
    $\{X_{S_i}(t_{k+1})\}_{1\le i\le d_S} \xleftarrow{(2.13)} \{\tilde{X}_{S_i}(t_{k+1}), x^b_{S_i}(t_k), n_{S_i}(x^b_{S_i}(t_k))\}_{1\le i\le d_S}$
    $X_E(t_{k+1}) \xleftarrow{(2.15)} \tilde{X}_E(t_{k+1}), x^b_E(t_k), n_E(x^b_E(t_k))$
    if $T_f<\infty$ **then**
      $\hat{p}(\mathcal{T}, X_S(\mathcal{T}), X_E(\mathcal{T}); \theta) \leftarrow p(t_{k+1}, X_S(t_{k+1}), X_E(t_{k+1}))$
      Break
    end if
  end for
  $\mathrm{Loss}(\theta) \leftarrow \mathbb{E}[|\hat{p}(\mathcal{T}, X_S(\mathcal{T}), X_E(\mathcal{T}); \theta) - g(\mathcal{T}, X_S(\mathcal{T}), X_E(\mathcal{T}))|^2]$
  $\theta \xleftarrow{\text{SGD algorithm}} \theta, \alpha, \nabla_\theta \mathrm{Loss}(\theta)$
  step $\leftarrow$ step $+1$
end while
return θ_p

$$
\begin{aligned}
Y(t) = Y(0) &- \int_0^t \mathbb{I}_{\{s\le\mathcal{T}\}} f(s, X_S(s), X_E(s), Y(s), Z_E(s))ds \\
&+ \sum_{i=1}^{d_S} \int_0^t \mathbb{I}_{\{s\le\mathcal{T}\}} [Z_{S_i}(s)]^T dB_{S_i}(s) + \int_0^t \mathbb{I}_{\{s\le\mathcal{T}\}} [Z_E(s)]^T dB_E(s) \\
&- \int_0^t \mathbb{I}_{\{s\le\mathcal{T}\}} h(s, X_S(s), X_E(s)) \left(\sum_{i=1}^{d_S} d|R_{S_i}|(s) + d|R_E|(s) \right).
\end{aligned}
$$

The solution of the FBSDEs (2.8) and (2.11) is a minimizer of the variational problem (2.21). The well-posedness of the FBSDEs ensures the existence and uniqueness of the minimizer and allows us to solve PDE (2.1) with complex constraints by solving the

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 10/27/26 to 161.81.191.158 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

variational problem (2.21), thereby avoiding automatic differentiation, the sampling of boundary points, global solution, and the singularity problem.

3. Numerical results. In this section, we will present the performance of the Constraint-Informed BSDE method through specific numerical examples, along with that of the baseline PINN [15]. The neural network employed is a fully connected neural network with 5 hidden layers, a width of $32+d+d_{\text{out}}$ with the input dimension d and the output dimension d_{out}, and the Tanh activation function. The Adam optimizer [9] is used to minimize the loss function, with the learning rate of 10^{-3}, the number of iterations 10^4, and the batch size 512. To validate the method, we consider the relative L_2 error

$$E_{L_2} = \frac{\sqrt{\sum_{i=1}^{M} |\hat{p}(x_i) - p(x_i)|^2}}{\sqrt{\sum_{i=1}^{M} |p(x_i)|^2}}, \tag{3.1}$$

where M represents the total number of the test points chosen randomly in the domain, and $\hat{p}(x_i)$ and $p(x_i)$ represent the predicted and the exact solution values, respectively. We open the source code for readers to test: https://github.com/WANGH950/CIBSDE.

3.1. Burgers' equation. We consider the Burgers' equation

$$\begin{cases} \dfrac{\partial}{\partial t}p(t,x) - p(t,x)\dfrac{\partial}{\partial x}p(t,x) + \dfrac{0.01}{\pi}\dfrac{\partial^2}{\partial x^2}p(t,x) = 0, & (t,x) \in [0,T] \times [-1,1], \\ p(t,-1) = p(t,1) = 0, & t \in [0,T], \\ p(T,x) = -\sin(\pi x), & x \in [-1,1]. \end{cases} \tag{3.2}$$

The solution $p(t,x)$ to (3.2) exhibits a sharp change near $x = 0$. Consequently, the baseline PINN incurs significant errors when using uniform sampling [5]. We adopt this case as a numerical example for nonlinear problems.

We set $T = 1$, $N = 100$, and obtain the solutions of (3.2) at the time instants $t = 0, 0.25, 0.5, 0.75$ using the Constraint-Informed BSDE method and the baseline PINN. Both methods are implemented with the Adam optimizer for 10^4 iterations, without any pretraining or fine-tuning.

Table 1 presents the predicted relative L_2 errors of two methods, and Figure 2 plots the solutions to (3.2) obtained by two deep learning methods at four different time steps $t = 0.75$, 0.5, 0.25, 0. The baseline PINN is trained 10^4 iterations with a learning rate of 10^{-3} by uniformly sampling batch size samples, respectively, within the domain $[0,1] \times [-1,1]$, on the boundary $[0,1] \times \{-1,1\}$, and at the terminal time $\{1\} \times [-1,1]$. It can be observed that as the predicted time horizon increases, the error of the Constraint-Informed BSDE method rises relatively slowly, while the error of baseline PINN gradually accumulates. Furthermore, the Constraint-Informed BSDE method can capture the sharp change features of the solution without any additional operations, whereas PINN performs slightly worse in this aspect. This also demonstrates that the Constraint-Informed BSDE method exhibits strong robustness.

3.2. Heat conduction in a square metal. We consider the 2-dimensional heat equation

$$\begin{cases} \dfrac{\partial}{\partial t}p(t,x) + D\Delta_x p(t,x) = 0, & (t,x) \in [0,T] \times \Omega, \\ \nabla_x p(t,x) \cdot n_x = h(x), & (t,x) \in [0,T] \times \partial\Omega, \\ p(T,x) = (\sin(x_1) + 1)(\cos(x_2) + 1), & x \in \Omega, \end{cases} \tag{3.3}$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 10/27/26 to 161.81.191.158 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

TABLE 1

The relative L_2 errors between the solutions of (3.2) *(obtained by Constraint-Informed BSDE method and baseline PINN) and the "exact" solution (obtained by the finite difference method) at four time instants $t = 0.75, 0.5, 0.25, 0$.*

	t	0.75	0.5	0.25	0
E_{L_2}	PINN	0.1112	0.3034	0.4384	0.5314
	Ours	**0.0171**	**0.0772**	**0.0628**	**0.0819**

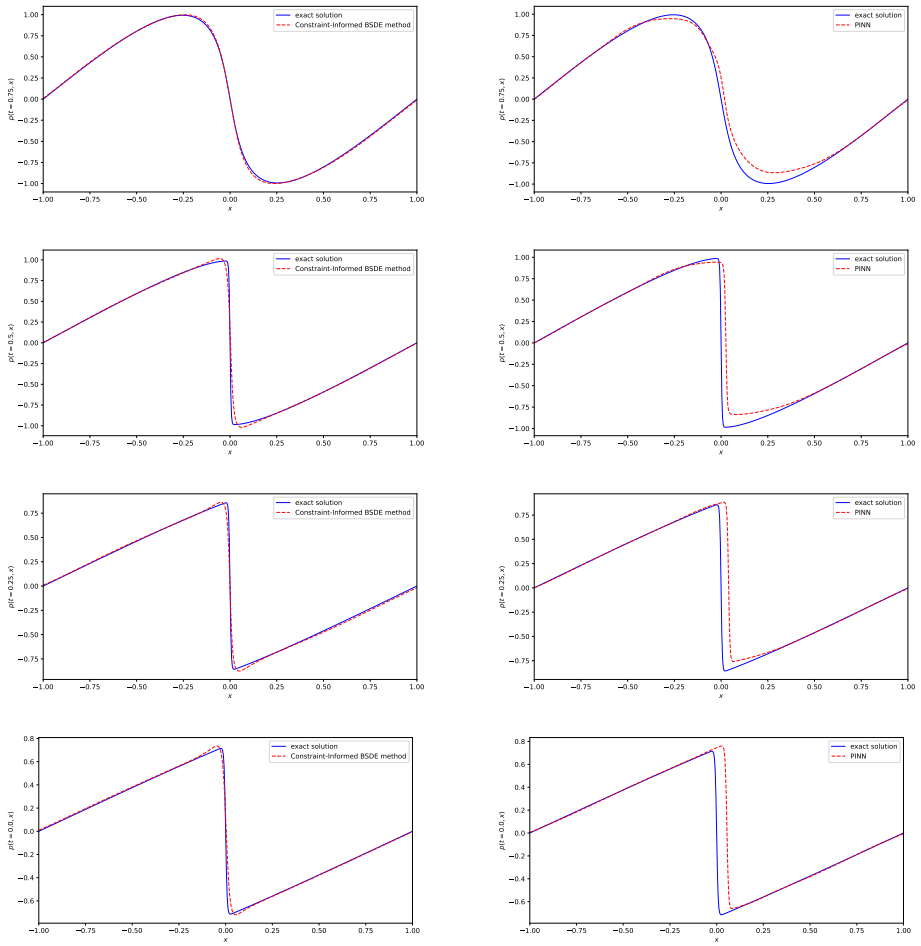


FIG. 2. *Plots of the solutions to* (3.2) *obtained by Constraint-Informed BSDE method (left column, red dashed line) and the baseline PINN (right column, red dashed line) at four different time steps $t = 0.75, 0.5, 0.25, 0$, where the "exact" solutions are obtained by the finite difference method.*

where $\Omega = [0, 2\pi] \times [0, 2\pi]$ is a square region and

$$h(x) = \begin{cases} 0, & x \in [0, 2\pi] \times \{0, 2\pi\}, \\ 1, & x \in \{0\} \times [0, 2\pi], \\ -1, & x \in \{2\pi\} \times [0, 2\pi]. \end{cases}$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

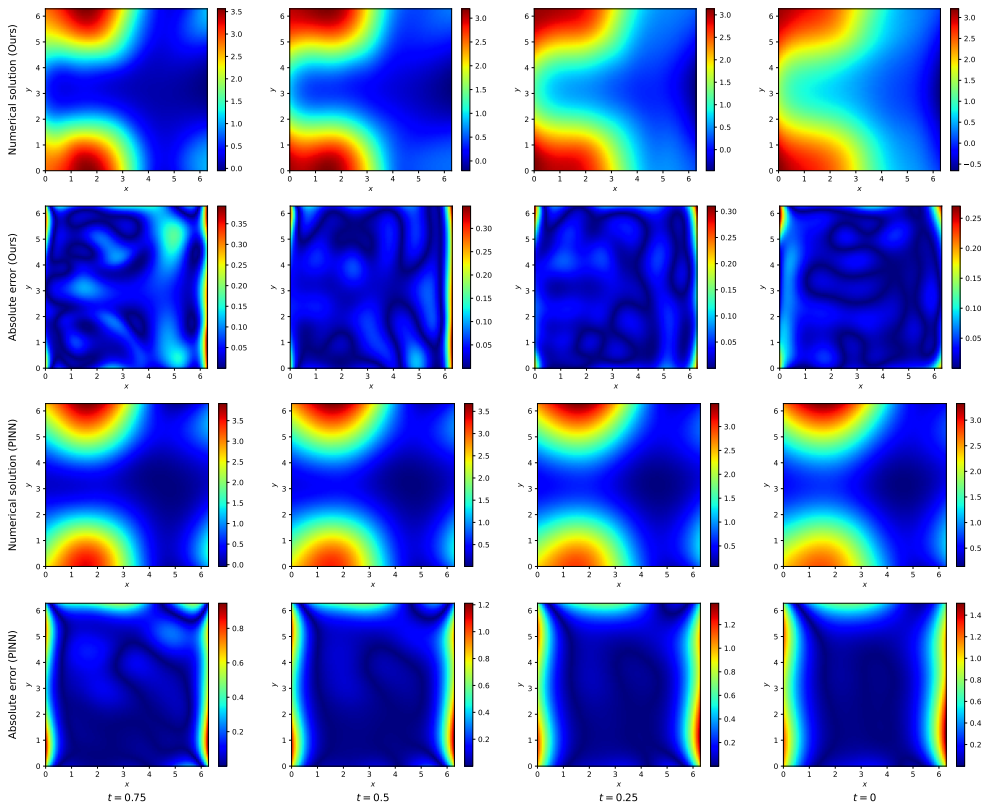


FIG. 3. *Plots of the solutions to* (3.3) *obtained by the Constraint-Informed BSDE method (first row) and the baseline PINN (third row) at time instants* $t = 0.75,\ 0.5,\ 0.25,\ 0$ *(from left column to right column), along with their absolute errors (the second line corresponds to the Constraint-Informed BSDE method and the fourth line is for the baseline PINN) relative to the "exact" solution (obtained by the finite difference method).*

The solution to (3.3) describes the temperature distribution on a square metal plate at time $T-t$ (with thermal diffusivity D and initial temperature distribution $p(T,x)$). During the evolution, a constant unit heat flux is applied to the left boundary to heat the plate, while a constant unit heat flux is released through the right boundary. The upper and lower boundaries are thermally insulated, meaning no heat is lost through them. Therefore, we use this example to test the performance of the Constraint-Informed BSDE method in handling problem with Neumann boundary condition.

We set $T = 1$, $D = 0.5$, $N = 100$, and obtain the solutions of (3.3) using the Constraint-Informed BSDE method and the baseline PINN. Figure 3 shows the solutions to (3.3) obtained by the Constraint-Informed BSDE method (first line) and the baseline PINN (third line) at time instants $t = 0.75,\ 0.5,\ 0.25,\ 0$ (from left column to right column), along with their absolute errors (the second line corresponds to the Constraint-Informed BSDE method and the fourth line is for the baseline PINN) relative to the "exact" solution (obtained by the finite difference method). Both methods are trained without any other operations, using the Adam optimizer with a learning rate of 10^{-3} and a batch size of 512, and 10^4 iteration steps. It can be observed that our algorithm significantly outperforms the baseline PINN.

Table 2 presents the relative L_2 errors of PINN and our algorithm at the four time instants. It can be seen that the relative error of the Constraint-Informed BSDE

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 10/27/26 to 161.81.191.158 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

TABLE 2

The relative L_2 errors between the solutions of (3.3) (obtained by Constraint-Informed BSDE method and baseline PINN), and the "exact" solution (obtained by the finite difference method) at four time instants $t = 0.75, 0.5, 0.25, 0$.

	t	0.75	0.5	0.25	0
E_{L_2}	PINN	0.1275	0.1915	0.2472	0.2963
	Ours	**0.0557**	**0.0415**	**0.0300**	**0.0266**

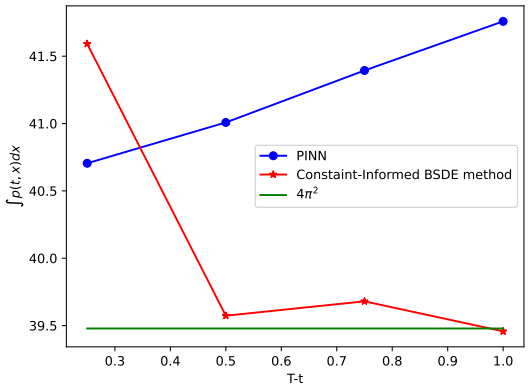


FIG. 4. *Plots of the variation of the energy over time, computed from the solutions obtained by the Constraint-Informed BSDE method (red solid line with stars) and the PINN method (blue solid line with dots), as well as the exact energy (green solid line).*

method remains basically stable, while the relative error of PINN accumulates over time.

Since (3.3) has identical heat influx and outflux, its solution must satisfy the conservation law. Figure 4 illustrates the energy variation over time obtained by the two methods. It is evident that the energy obtained by the Constraint-Informed BSDE method remains stable around the exact energy, whereas the baseline PINN exhibits an unstable linear increasing trend.

3.3. Diffusion in an annular region. We consider the 2-dimensional diffusion equation

$$
\begin{cases}
\dfrac{\partial}{\partial t}p(t,x) + D\Delta_x p(t,x) = 0, & (t,x) \in [0,T] \times \Omega, \\
\nabla_x p(t,x) \cdot n_x = 0, & (t,x) \in [0,T] \times \partial\Omega, \\
p(T,x) = \dfrac{16}{\pi}, & x \in B(x_0,r), \\
p(T,x) = 0, & x \in \Omega \setminus B(x_0,r),
\end{cases} \tag{3.4}
$$

where $\Omega = B(0,1) \setminus B(0,0.5)$ is an annular region, and $x_0 = [0.75, 0]$, $r = 0.25$. The solution to (3.4) describes the distribution at time $T - t$ of a freely moving Brownian particle (with a diffusion coefficient D and the initial distribution $p(T,x)$) within the domain Ω. The outer boundary $\Gamma_1 = \{x : |x| = 1\} \subset \partial\Omega$ is convex outward, and its inner boundary $\Gamma_2 = \{x : |x| = 0.5\} \subset \partial\Omega$ is concave inward. This example is used to test the performance of the Constraint-Informed BSDE method in solving diffusion equations over complex domains.

We set $T = 1$, $D = 0.5$, $N = 100$, and obtain the solutions of (3.4) using the Constraint-Informed BSDE method and the baseline PINN. Figure 5 plots the

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 10/27/26 to 161.81.191.158 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

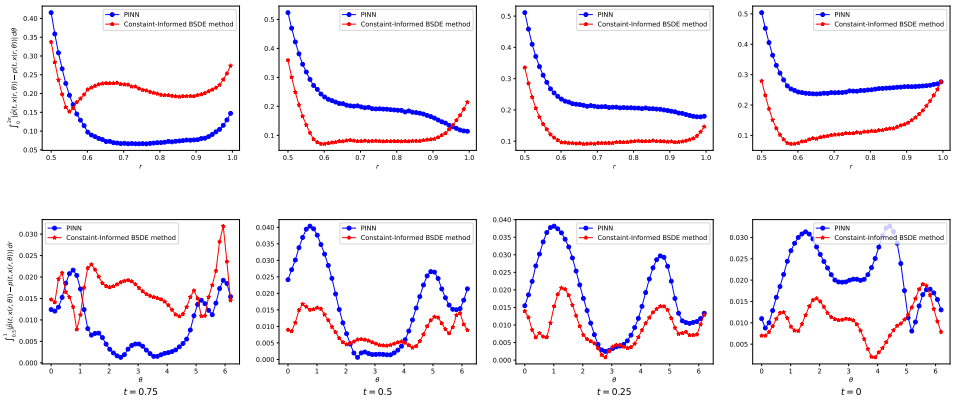


FIG. 5. *Plots of the absolute error of the solution to* (3.4)*, obtained by the Constraint-Informed BSDE method (red solid line with stars) and the baseline PINN method (blue solid line with dots). From left to right, the plots correspond to four different time points:* $t = 0.75, 0.5, 0.25, 0$*. The first row shows the results of the absolute error integrated with respect to the angle* θ*, while the second row presents the results of the absolute error integrated with respect to the radius* r*.*

TABLE 3

The relative L_2 *errors between the solutions of* (3.4) *(obtained by Constraint-Informed BSDE method and baseline PINN), and the "exact" solution (obtained by approximating the probability density through simulating* 10^9 *trajectories using the Monte Carlo method) at four time instants* $t = 0.75, 0.5, 0.25, 0$*.*

	t	0.75	0.5	0.25	0
E_{L_2}	PINN	**0.0422**	0.0798	0.0880	0.0964
	Ours	0.0553	**0.0428**	**0.0483**	**0.0525**

absolute errors of the solutions to (3.4) relative to the "exact" solutions (obtained by approximating the probability density through the independent simulation of 10^9 trajectories via the Monte Carlo method) obtained by the Constraint-Informed BSDE method (red solid line with stars) and the baseline PINN method (blue solid line with dots) at time instants $t = 0.75, 0.5, 0.25, 0$ (from left column to right column). Both methods are trained without any other operations, using the Adam optimizer with a learning rate of 10^{-3} and a batch size of 512, and 10^4 iteration steps. It can be shown that the absolute errors of the Constraint-Informed BSDE method are comprehensively smaller than that of the solutions obtained by the baseline PINN method. Table 3 presents the relative L_2 errors of the baseline PINN method and our proposed Constraint-Informed BSDE algorithm at the four aforementioned time instants, from which it can be can be observed that the relative L_2 error of the latter remains basically stable at the order of $\mathcal{O}(10^{-2})$, whereas the relative L_2 error of the former not only lies at the order of $\mathcal{O}(10^{-1})$ but also accumulates continuously over time.

3.4. Diffusion on a sphere. We consider the diffusion equation on a sphere, which is expressed as

$$\text{(3.5)} \quad \begin{cases} \dfrac{\partial}{\partial t} p(t, x(\theta,\phi)) + D\Delta_S p(t, x(\theta,\phi)) = 0, & (t,\theta,\phi) \in [0,T] \times [0,\pi] \times [0,2\pi], \\ p(T, x(\theta,\phi)) = p_0(x(\theta,\phi)), & (\theta,\phi) \in \times [0,\pi] \times [0,2\pi], \end{cases}$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 10/27/26 to 161.81.191.158 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

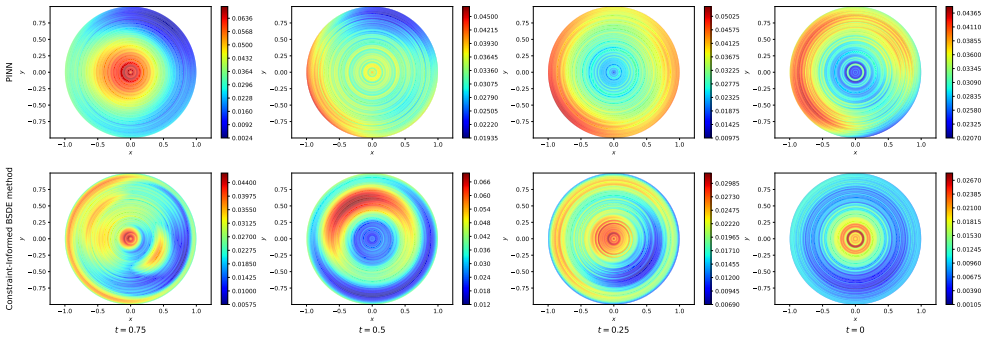


FIG. 6. *Plots the top views of the absolute errors (obtained by summing the absolute errors at symmetric points between the northern and southern hemispheres) of the solutions to* (3.5) *obtained by the baseline PINN method (first row) and the Constraint-Informed BSDE method (second row) at time instants* $t = 0.75,\ 0.5,\ 0.25,\ 0$ *(from left column to right column), relative to the "exact" solutions. The "exact" solutions are got by approximating the probability density through simulating* 10^8 *trajectories using the Monte Carlo method.*

where D is the diffusion coefficient and Δ_S is the Laplace–Beltrami operator on the sphere. The solution to (3.5) represents the distribution of the Brownian particles undergoing free motion on the unit sphere. Among the components involved, the Laplace–Beltrami operator exhibits singularity (when $\theta \to 0,\ \pi$); however, this singularity is a "nonphysical" characteristic. Thus, we utilize this example to test the performance of the Constraint-Informed BSDE method under manifold constraints.

We set $T = 1,\ D = 0.5,\ N = 100,$

$$p_0(x(\theta,\phi)) = \begin{cases} \dfrac{1}{\sin\theta}, & \dfrac{\pi}{4} \le \theta \le \dfrac{3\pi}{4}, \\ 0 & \text{otherwise}, \end{cases}$$

and obtain the solutions of (3.5) using the Constraint-Informed BSDE method and the baseline PINN. Figure 6 shows the top views of the absolute errors (obtained by summing the absolute errors at symmetric points between the northern and southern hemispheres) of the solutions to (3.5) obtained by the baseline PINN method (first row) and the Constraint-Informed BSDE method (second row) at time instants $t = 0.75,\ 0.5,\ 0.25,\ 0$ (from left column to right column), relative to the "exact" solutions. The "exact" solutions are got by approximating the probability density through simulating 10^8 trajectories using the Monte Carlo method. Both methods are trained without any other operations, using the Adam optimizer with a learning rate of 10^{-3} and a batch size of 512, and 10^4 iteration steps. It can be observed that our algorithm significantly outperforms the baseline PINN.

Table 4 presents the relative L_2 errors of PINN and the Constraint-Informed BSDE method at the four time instants. It can be observed that the relative L_2 errors of these two methods remain basically stable, while the error of the Constraint-Informed BSDE method is smaller.

3.5. Multiple microtubule search. We consider the search model of P microtubules, and the corresponding macroscopic equation can be expressed as

$$\frac{\partial}{\partial t} p(t, x(\theta_1,\phi_1), \ldots, x(\theta_P,\phi_P)) + \sum_{i=1}^{P} \frac{D}{r_i^2} \Delta_{S_i} p(t, x(\theta_1,\phi_1), \ldots, x(\theta_P,\phi_P)) = 0 \tag{3.6}$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 10/27/26 to 161.81.191.158 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

Downloaded 10/27/26 to 161.81.191.158 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

TABLE 4

The relative L_2 errors between the solutions of (3.5) (obtained by Constraint-Informed BSDE method and baseline PINN), and the "exact" solution (obtained by the Monte Carlo method) at four time instants $t = 0.75,\ 0.5,\ 0.25,\ 0$.

	t	0.75	0.5	0.25	0
E_{L_2}	PINN	0.0291	**0.0228**	0.0234	0.0214
	Ours	**0.0221**	0.0262	**0.0144**	**0.0084**

with the Neumann boundary conditions $\nabla_{x_i} p \cdot n_i|_{x(\theta_i,\phi_i)\in\partial\Omega_i} = 0,\ i = 1,\dots,P$, and the terminal condition $p|_{t=T} = p_0$. This model describes the process of P independently rotating microtubules with respective lengths $r_i,\ i = 1,\dots,P$, searching for chromosomes within a cell [18], and can also describe P independent Brownian particles diffuse on the unit sphere with different diffusion coefficients $D_i,\ i = 1,\dots,P$. Due to the constraint of cell volume, the microtubules undergo mirror reflection upon contacting the cell membrane. In this model, the cell is treated as a sphere $B([0,0,R]^T, R)$ with center $[0,0,R]^T$ and radius R, and the microtubules rotate around the origin. Since microtubules of different lengths correspond to different polar angle ranges, their search regions and reflective boundaries are also distinct (here, $\Omega_i = \{x(\theta_i,\phi_i); 0 \le \theta_i < \arccos(r_i/2R), 0 \le \phi_i < 2\pi\}$). A single cell typically contains multiple microtubules, so this model is a high-dimensional problem. Thus, we test the performance of the Constraint-Informed BSDE method in solving problems involving complex domains with reflective boundary conditions by using this example.

Firstly, to test the performance of the Constraint-Informed BSDE method in solving problems with Neumann boundary conditions on the sphere, we consider a low-dimensional case. Therefore, we set $P = 1$, $T = 1$, $D = 0.1$, $r_1 = 1$, $R = 2$, $N = 100$, and

$$p_0(x(\theta,\phi)) = \begin{cases} 1, & \theta < \dfrac{\pi}{4}, \\ 0 & \text{otherwise.} \end{cases}$$

Figure 7 illustrates the performance of PINN and the Constraint-Informed BSDE method on this problem. The "exact" solution is obtained via the Monte Carlo method, which approximates the expectation using the sample mean by stochastically simulating 10^7 independent particle trajectories to approach the stochastic representation given in (2.10). It can be observed that the errors obtained by the two methods are basically the same.

Figure 8 shows the relationship between the L_2 relative error (with respect to (3.6), compared with the "exact" solution obtained by Monte Carlo method) and the number of iterations for the two deep learning methods in low-dimensional case ($P = 1$). It is evident that both methods converge rapidly, while the Constraint-Informed BSDE method exhibits greater stability.

Then, we aim to the high-dimensional case $P = 3$. We set $T = 1$, $D = 0.1$, $r_1 = 1$, $r_2 = 1.3$, $r_3 = 1.5$, $R = 2$, $N = 100$, and

$$p_0(x(\theta_1,\phi_1), x(\theta_2,\phi_2), x(\theta_3,\phi_3)) = \begin{cases} 1, & \theta_i < \dfrac{\pi}{4}, i = 1,2,3, \\ 0 & \text{otherwise.} \end{cases}$$

We focus on the solution $p(t = 0, x_1 = [0,0,1]^T, x_2 = [0,0,1]^T, x(\theta_3,\phi_3))$. For the Constraint-Informed BSDE method, benefiting from the existence and uniqueness theory of BSDE, we only need to generate samples with initial values within our interest region. In contrast, for PINN, it is necessary to generate both global samples

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

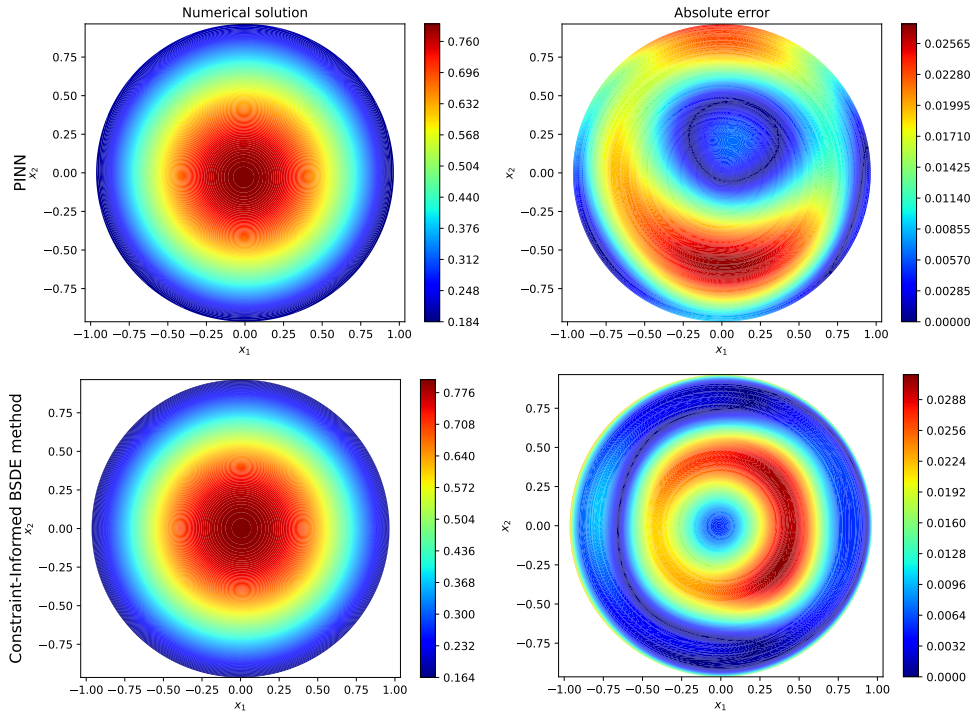


FIG. 7. *Plots the top view of the solutions (left column) to* (3.6) *for low-dimensional case* ($P = 1$) *obtained by the baseline PINN (first row), the Constraint-Informed BSDE method (second row), and their absolute errors compared with the exact solutions (right column, obtained by Monte Carlo method).*

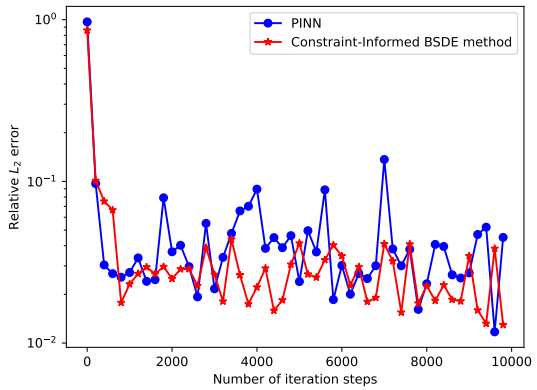


FIG. 8. *Plots of the relationship between the L_2 relative error (with respect to* (3.6)*, compared with the "exact" solution obtained by Monte Carlo method) and the number of iterations for the Constraint-Informed BSDE method (red solid line with stars) and the PINN method (blue solid line with dots) in low-dimensional case* ($P = 1$).

and boundary samples, along with fine-tuning of the corresponding parameters. Figure 9 shows the performance of PINN and the Constraint-Informed BSDE method on this high-dimensional problem. The "exact" solution is obtained via the Monte Carlo method, which approximates the expectation using the sample mean by

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 10/27/26 to 161.81.191.158 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

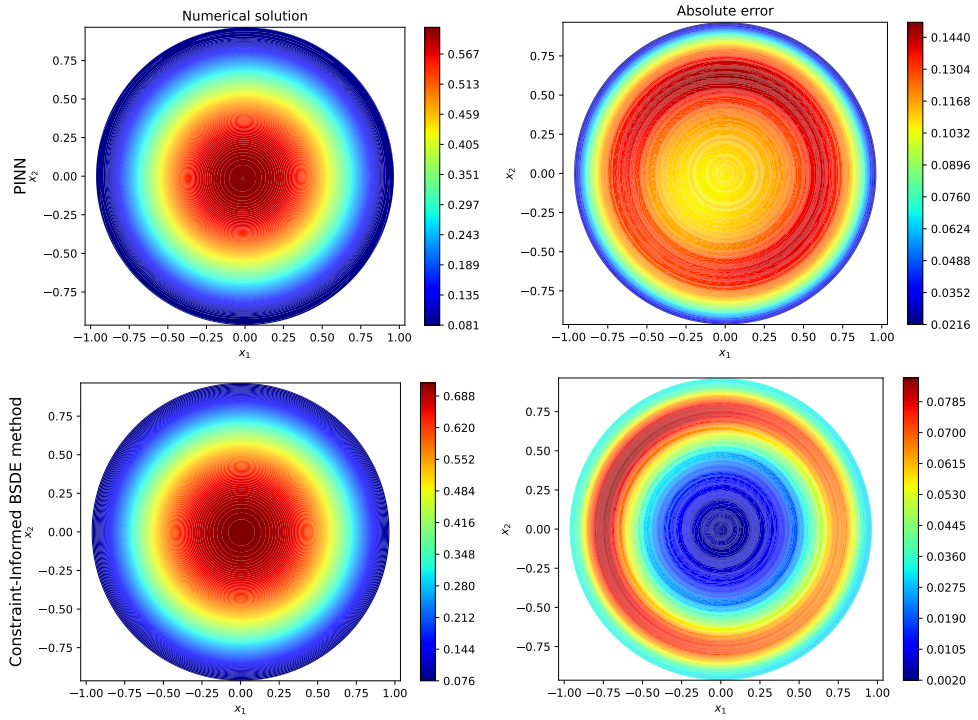


FIG. 9. *Plots the top view of the solutions (left column) to* (3.6) *for high-dimensional case* ($P = 3$) *obtained by the baseline PINN (first row), the Constraint-Informed BSDE method (second row), and their absolute errors compared with the exact solutions (right column, obtained by Monte Carlo method).*

stochastically simulating 10^5 independent particle trajectories to approach the stochastic representation given in (2.10). It can also be observed that the results obtained by PINN are slightly inferior to those of the Constraint-Informed BSDE method.

Figure 10 shows the relationship between the L_2 relative error (with respect to (3.6), compared with the "exact" solution obtained by Monte Carlo method) and the number of iterations for the two deep learning methods in high-dimensional case ($P = 3$). It can be observed that the Constraint-Informed BSDE method converges rapidly, exhibits higher accuracy than the baseline PINN, and demonstrates strong stability.

3.6. Multiple microtubules capture a kinetochore. As the primary method of cell division, mitosis can be divided into five stages: prophase, prometaphase, metaphase, anaphase, and telophase. Among these, the mechanism controlling the distribution of chromosomes into daughter cells is crucial. Errors in this distribution may lead to a range of diseases, such as phenotypic defects, tumors, and aging [10]. Mitosis primarily ensures accurate chromosome segregation during anaphase through the capture of kinetochores by microtubules, a process that mainly occurs during prometaphase [8]. The movement of the tip of a microtubule with a fixed length can be regarded as a normal rotational search process around the centrosome, while kinetochores undergo normal free movement within the cell [18]. We consider a model where P microtubules capture a single kinetochore, and its corresponding macroscopic equation can be expressed as

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 10/27/26 to 161.81.191.158 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

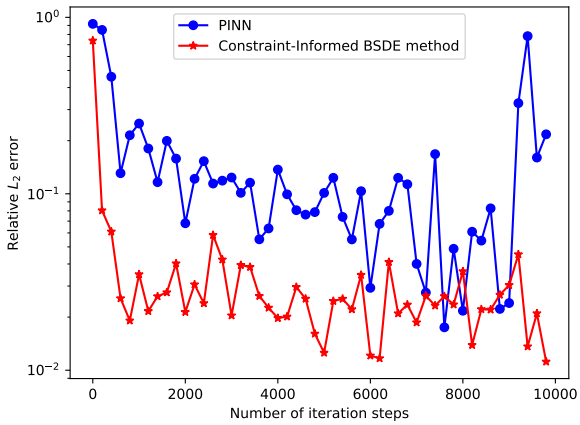


FIG. 10. *Plots of the relationship between the L_2 relative error (with respect to* (3.6)*, compared with the "exact" solution obtained by Monte Carlo method) and the number of iterations for the Constraint-Informed BSDE method (red solid line with stars) and the PINN method (blue solid line with dots) in high-dimensional case ($P = 3$).*

$$\begin{aligned}(3.7)\qquad &\frac{\partial}{\partial t}p(t,x(\theta_1,\phi_1),\dots,x(\theta_P,\phi_P),y)+D_y\Delta_y p(t,x(\theta_1,\phi_1),\dots,x(\theta_P,\phi_P),y)\\ &+\sum_{i=1}^{P}\frac{D_x}{r_i^2}\Delta_{S_i}p(t,x(\theta_1,\phi_1),\dots,x(\theta_P,\phi_P),y)=0\end{aligned}$$

with the Neumann boundary conditions $\nabla_{x_i}p\cdot n_i|_{(\theta_i,\phi_i)\in\partial\Omega_i}=0$, $i=1,\dots,M$, and $\nabla_y p\cdot n_y|_{y\in\partial B}=0$, the Dirichlet boundary condition $p|_{|x(\theta_i,\phi_i)-y|<a}=0$, and the terminal condition $p|_{t=T}=p_0$. Here, $x(\theta_i,\phi_i)$ denotes the rotational position of the ith microtubule, y is a 3-dimensional space variable, representing the position of the kinetochore, $B=B([0,0,R]^T,R)$ represents the cellular region, $\Omega_i=\{x(\theta_i,\phi_i);0\le\theta_i<\arccos(r_i/2R),0\le\phi_i<2\pi\}$ denotes the rotational region of the ith microtubule, and the solution p represents the number of cells (or cell density, or probability) where the tip of the ith microtubule is at position $x(\theta_i,\phi_i)$ and the kinetochore is at position y. Since the entire cell disappears as soon as any microtubule captures the target kinetochore (absorbing boundary condition), this model describes the survival probability density of a kinetochore captured by P microtubules [18]. Owing to the constraint of cell volume, both the microtubule tips and kinetochores will rebound back into the sphere via mirror reflection upon contacting the boundary. Due to the complexity of the boundaries in this high-dimensional problem, it is extremely difficult to perform sampling for the stable training of PINN; therefore, we use this example to test the performance of the Constraint-Informed BSDE method in solving high-dimensional problems with mixed boundary conditions.

We consider the scenario where a single kinetochore is being captured by four microtubules, i.e., $P=4$. Therefore, we set $D_x=0.5$, $D_y=1$, $r_1=1$, $r_2=1.3$, $r_3=1.5$, $r_4=1.8$, $a=0.1$, $T=1$, $N=100$,

$$p_0(x(\theta_1,\phi_1),\dots,x(\theta_4,\phi_4),y)=\begin{cases}1, & 2.1<y_3,\theta_i<\dfrac{\pi}{4},i=1,\dots,4,\\ 0 & \text{otherwise},\end{cases}$$

and a learning rate of 5×10^{-4}. Figure 11 shows the variation of the solution to (3.7) (obtained by the Constraint-Informed BSDE method) with the number of iterations

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 10/27/26 to 161.81.191.158 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

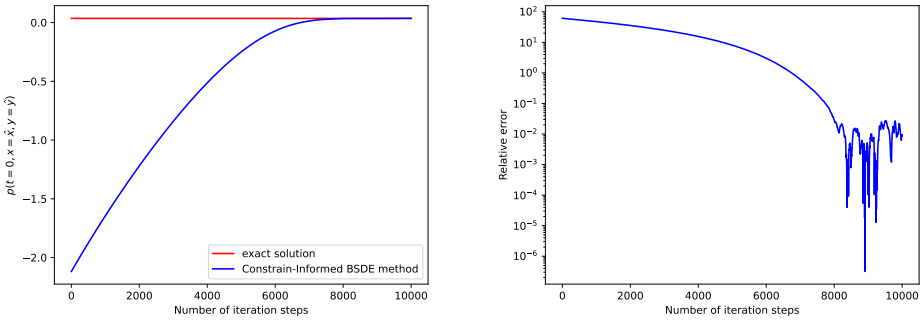


FIG. 11. *Plots of the variation of the solution to* (3.7) *(obtained by the Constraint-Informed BSDE method) with the number of iterations at the fixed position* $t = 0$, $\hat{x} = [0,0,1,0,0,1,0,0,1,0,0,1]^T$, $\hat{y} = [0,0,1]^T$ *(left), as well as the variation of its relative error (right).*

at the fixed position $t = 0$, $\hat{x} = [0,0,1,0,0,1,0,0,1,0,0,1]^T$, $\hat{y} = [0,0,1]^T$ (left), as well as the variation of its relative error (right); it can be observed that the deep learning results converge after 10^3 iterations and stabilize around the exact solution, where the exact solution 0.035158 is obtained by independently sampling 10^9 particles via the Monte Carlo method and approximating stochastic representation (2.10) using the sample mean.

4. Conclusion. In this paper, based on the physical meaning of boundary conditions, we derive the BSDEs that the solutions to PDEs with complex mixed boundary conditions (Neumann and Dirichlet boundary conditions) on a sphere satisfy, which comply with the existence and uniqueness theory of BSDEs with stopping time [14, 11, 2, 13, 12]. Furthermore, we develop a corresponding deep learning algorithm, named as the Constraint-Informed BSDE method. A major advantage of the Constraint-Informed BSDE method is that it does not require considering the sampling of boundary points and terminal time (corresponding to the initial time in the original problem) by training neural networks to get the solution at any attention point without requiring a whole solution to be computed. Thus, it is more consistent with "physics-informed" deep learning method. Finally, extensive numerical experiments demonstrate that, without any additional fine-tuning operations, deep Constraint-Informed BSDE method is more convenient, more accurate, and more stable than the baseline PINN when solving problems with complex boundary conditions. Deep Constraint-Informed BSDE method can be extended to solve PDEs on general manifolds and problems in other complex domains, which will be our future work. Owing to the connection between BSDEs and the stochastic microscopic models used in deriving the equations [1, 17], we will also explore algorithms for complex domains that integrate data and mechanism in our future work.

Appendix A. Derivation of (2.9). For each $0 \le t \le \mathcal{T}$, we have

$$\begin{aligned}
&p(t, X_S(t), X_E(t)) - p(0, X_S(0), X_E(0)) \\
&\quad = \sum_{0 \le s < t} \left[p(s + ds, X_S(s + ds), X_E(s + ds)) - p(s, X_S(s), X_E(s)) \right] \\
&\quad = \sum_{0 \le s < t} \mathbb{I}_{\{(\tilde{X}_S(s+ds), \tilde{X}_E(s+ds)) \in \overline{\Omega}\}} \left[p(s + ds, X_S(s + ds), X_E(s + ds)) \right. \\
&\qquad \left. - p(s, X_S(s), X_E(s)) \right]
\end{aligned}$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 10/27/26 to 161.81.191.158 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

$$
\begin{aligned}
&+\sum_{0\le s<t}\mathbb{I}_{\{(\tilde{X}_S(s+ds),\tilde{X}_E(s+ds))\notin\overline{\Omega}\}}\left[p(s+ds,X_S(s+ds),X_E(s+ds))\right.\\
&\left.-p(s,X_S(s),X_E(s))\right]\\
&=\mathrm{I}+\mathrm{II}.
\end{aligned}
$$

We first assume $d_S=1$. For the first term I, by using definitions (2.4)–(2.7), property (2.3), and Itô's formula, one can get

$$
\begin{aligned}
&p(s+ds,X_S(s+ds),X_E(s+ds))-p(s,X_S(s),X_E(s))\\
&\quad=p(s+ds,T(X_{S_1}(s))^{-1}x(\pi/2+\sqrt{2D_1}dB_{\theta_1}(s),\sqrt{2D_1}dB_{\phi_1}(s)),X_E(s+ds))\\
&\qquad-p(s,T(X_{S_1}(s))^{-1}x(\pi/2,0),X_E(s))\\
&\quad=p_{X_{S_1}(s)}(s+ds,x(\pi/2+\sqrt{2D_1}dB_{\theta_1}(s),\sqrt{2D_1}dB_{\phi_1}(s)),X_E(s+ds))\\
&\qquad-p_{X_{S_1}(s)}(s,x(\pi/2,0),X_E(s))\\
&\quad=-f\left(s,X_S(s),X_E(s),p(s,X_S(s),X_E(s)),\left(\sigma^T\nabla_E p\right)(s,X_S(s),X_E(s))\right)ds\\
&\qquad+\sqrt{2D_1}[\nabla_{\theta,\phi}p_{X_{S_1}(s)}(s,x(\pi/2,0),X_E(s))]^T dB_{S_1}(s)\\
&\qquad+\nabla_E p_{X_{S_1}(s)}(s,x(\pi/2,0),X_E(s))\sigma(s,X_E(s))dB_E(s)\\
&\quad=-f\left(s,X_S(s),X_E(s),p(s,X_S(s),X_E(s)),\left(\sigma^T\nabla_E p\right)(s,X_S(s),X_E(s))\right)ds\\
&\qquad+\sqrt{2D_1}[\nabla_{\theta,\phi}p(s,T(X_{S_1}(s))^{-1}x(\pi/2,0),X_E(s))]^T dB_{S_1}(s)\\
&\qquad+\nabla_E p(s,T(X_{S_1}(s))^{-1}x(\pi/2,0),X_E(s))\sigma(s,X_E(s))dB_E(s)\\
&\quad=-f\left(s,X_S(s),X_E(s),p(s,X_S(s),X_E(s)),\left(\sigma^T\nabla_E p\right)(s,X_S(s),X_E(s))\right)ds\\
&\qquad+\sqrt{2D_1}[\nabla_{S_1}p(s,X_S(s),X_E(s))]^T T(X_{S_1}(s))^{-1}\nabla_{\theta,\phi}(\pi/2,0)dB_{S_1}(s)\\
&\qquad+\nabla_E p(s,X_S(s),X_E(s))\sigma(s,X_E(s))dB_E(s).
\end{aligned}
$$

For the second term II, by using definitions (2.4)–(2.7), the Neumann boundary condition $\frac{\partial}{\partial n}p(t,x_S,x_E)|_{(x_S,x_E)\in\Gamma_1}=h(t,x_S,x_E)$, and following a derivation similar to the above, one can get

$$
\begin{aligned}
&p(s+ds,X_S(s+ds),X_E(s+ds))-p(s,X_S(s),X_E(s))\\
&\quad=-f\left(s,X_S(s),X_E(s),p(s,X_S(s),X_E(s)),\left(\sigma^T\nabla_E p\right)(s,X_S(s),X_E(s))\right)ds\\
&\qquad+\sqrt{2D_1}[\nabla_{S_1}p(s,X_S(s),X_E(s))]^T T(X_{S_1}(s))^{-1}\nabla_{\theta,\phi}(\pi/2,0)dB_{S_1}(s)\\
&\qquad+\nabla_E p(s,X_S(s),X_E(s))\sigma(s,X_E(s))dB_E(s)\\
&\qquad+\nabla_{S_1}p(s,X_S(s),X_E(s))\cdot n_{S_1}(X_{S_1}(s))d|R_{S_1}|(s)\\
&\qquad+\nabla_E p(s,X_S(s),X_E(s))\cdot n_E(X_E(s))d|R_E|(s)\\
&\quad=-f\left(s,X_S(s),X_E(s),p(s,X_S(s),X_E(s)),\left(\sigma^T\nabla_E p\right)(s,X_S(s),X_E(s))\right)ds\\
&\qquad+\sqrt{2D_1}[\nabla_{S_1}p(s,X_S(s),X_E(s))]^T T(X_{S_1}(s))^{-1}\nabla_{\theta,\phi}(\pi/2,0)dB_{S_1}(s)\\
&\qquad+\nabla_E p(s,X_S(s),X_E(s))\sigma(s,X_E(s))dB_E(s)\\
&\qquad-h(s,X_S(s),X_E(s))\left(d|R_{S_1}|(s)+d|R_E|(s)\right),
\end{aligned}
$$

where $d|R_{S_i}|(s)$ and $d|R_E|(s)$ are defined in section 2.1. Then, for any d_S, following the same derivation as above, one can obtain

$$
\begin{aligned}
&p(t,X_S(t),X_E(t))-p(0,X_S(0),X_E(0))\\
&\quad=-\int_0^t f\left(s,X_S(s),X_E(s),p(s,X_S(s),X_E(s)),\left(\sigma^T\nabla_E p\right)(s,X_S(s),X_E(s))\right)ds
\end{aligned}
$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 10/27/26 to 161.81.191.158 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

$$+\int_0^t \sqrt{2D_1}[\nabla_{S_1} p(s, X_S(s), X_E(s))]^T T(X_{S_1}(s))^{-1} \nabla_{\theta,\phi}(\pi/2, 0) dB_{S_1}(s)$$
$$+\int_0^t \nabla_E p(s, X_S(s), X_E(s)) \sigma(s, X_E(s)) dB_E(s)$$
$$-\int_0^t h(s, X_S(s), X_E(s)) \left(\sum_{i=1}^{d_S} d|R_{S_i}|(s) + d|R_E|(s) \right).$$

Finally, by adjusting the integral interval to $[t \wedge \mathcal{T}, \mathcal{T}]$, one can get (2.9).

Reproducibility of computational results. This paper has been awarded the "SIAM Reproducibility Badge: Code and data available" as a recognition that the authors have followed reproducibility principles valued by SISC and the scientific computing community. Code and data that allow readers to reproduce the results in this paper are available at https://github.com/WANGH950/CIBSDE and in the supplementary materials (M182682_Supplementary_Materials.zip [local/web 82.5MB]).

REFERENCES

[1] K. Arif, G. J. Xi, H. Wang, and W. H. Deng, *Deep learning for high-dimensional PDEs with fait-tailed Lévy measure*, J. Comput. Phys., 541 (2025), 114327, https://doi.org/10.1016/j.jcp.2025.114327.

[2] Z. Chen, *Existence and uniqueness for BSDE with stopping time*, Chin. Sci. Bull., 43 (1998), pp. 96–99, https://doi.org/10.1007/BF02883914.

[3] W. E and B. Yu, *The deep Ritz method: A deep learning-based numerical algorithm for solving variational problems*, Commun. Math. Stat., 6 (2018), pp. 1–12, https://doi.org/10.1007/s40304-018-0127-z.

[4] 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 (2023), pp. 1290–1303, https://doi.org/10.1137/22M1539952.

[5] Z. Gao, L. Yan, and T. Zhou, *Failure-informed adaptive sampling for PINNs*, SIAM J. Sci. Comput., 45 (2023), pp. A1971–A1994, https://doi.org/10.1137/22M1527763.

[6] J. Han, A. Jentzen, and W. E, *Solving high-dimensional partial differential equations using deep learning*, Proc. Natl. Acad. Sci. USA, 115 (2018), pp. 8505–8510, https://doi.org/10.1073/pnas.1718942115.

[7] J. Han and J. Long, *Convergence of the deep BSDE method for coupled FBSDEs*, Probab. Uncertain. Quant. Risk, 5 (2020), 5, https://doi.org/10.1186/s41546-020-00047-w.

[8] I. Kalinina, A. Nandi, P. Delivani, M. R. Chacón, A. H. Klemm, D. Ramunno-Johnson, A. Krull, B. Lindner, N. Pavin, and I. M. Tolić-Nørrelykke, *Pivoting of microtubules around the spindle pole accelerates kinetochore capture*, Nat. Cell Biol., 15 (2013), pp. 82–87, https://doi.org/10.1038/ncb2640.

[9] D. P. Kingma and J. Ba, *Adam: A Method for Stochastic Optimization*, in International Conference on Learning Representations (ICLR), 2015, https://doi.org/10.48550/arXiv.1412.6980.

[10] R. Li and J. Zhu, *Effects of aneuploidy on cell behaviour and function*, Nat. Rev. Mol. Cell Biol., 23 (2022), pp. 250–265, https://doi.org/10.1038/s41580-021-00436-9.

[11] E. Pardoux and S. Peng, *Backward stochastic differential equations and quasilinear parabolic partial differential equations*, in Stochastic Partial Differential Equations and Their Applications, B. L. Rozovskii and R. B. Sowers, eds., Springer, Berlin, Heidelberg, 1992, pp. 200–217.

[12] E. Pardoux and A. Rascanu, *Stochastic Differential Equations, Backward SDEs, Partial Differential Equations*, Springer, Cham, 2014, https://doi.org/10.1007/978-3-319-05714-9.

[13] E. Pardoux and S. Zhang, *Generalized BSDEs and nonlinear Neumann boundary value problems*, Probab. Theory Related Fields, 110 (1998), pp. 535–558, https://doi.org/10.1007/s004400050158.

[14] S. Peng, *Probabilistic interpretation for systems of quasilinear parabolic partial differential equations*, Stoch. Stoch. Rep., 37 (1991), pp. 61–74, https://doi.org/10.1080/17442509108833727.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 10/27/26 to 161.81.191.158 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

[15] 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*, J. Comput. Phys., 378 (2019), pp. 686–707, https://doi.org/10.1016/j.jcp.2018.10.045.

[16] H. Wang and W. H. Deng, *Solving bivariate kinetic equations for polymer diffusion using deep learning*, J. Mach. Learn., 3 (2024), pp. 215–244, https://doi.org/10.4208/jml.240124.

[17] H. Wang, X. H. Li, L. J. Zhao, and W. H. Deng, *Multiscale modeling and simulation for anomalous and nonergodic dynamics: From statistics to mathematics*, Fundam. Res., 6 (2026), pp. 659–671, https://doi.org/10.1016/j.fmre.2024.12.024.

[18] H. Wang, Y. Y. Xue, and W. H. Deng, *Feynman-Kac equation for microtubule search process in prometaphase*, Chaos, 36 (2026), 033145, https://doi.org/10.1063/5.0305406.

[19] Y. Zang, G. Bao, X. Ye, and H. Zhou, *Weak adversarial networks for high-dimensional partial differential equations*, J. Comput. Phys., 411 (2020), 109409, https://doi.org/10.1016/j.jcp.2020.109409.

Downloaded 10/27/26 to 161.81.191.158 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.