AIMS Mathematics, 9(7): 18688–18711.
DOI: 10.3934/math.2024909
Received: 15 March 2024
Revised: 22 May 2024
Accepted: 23 May 2024
Published: 04 June 2024

https://www.aimspress.com/journal/Math

Research article

Simulating a strongly nonlinear backward stochastic partial differential equation via efficient approximation and machine learning

Wanyang Dai*

Department of Mathematics and State Key Laboratory of Novel Software Technology, Nanjing University, Nanjing 210093, China

* **Correspondence:** Email: nan5lu8@nju.edu.cn.

Abstract: We have studied a strongly nonlinear backward stochastic partial differential equation (B-SPDE) through an approximation method and with machine learning (ML)-based Monte Carlo simulation. This equation is well-known and was previously derived from studies in finance. However, how to analyze and solve this equation has remained a problem for quite a long time. The main difficulty is due to the singularity of the B-SPDE since it is a strongly nonlinear one. Therefore, by introducing new truncation operators and integrating the machine learning technique into the platform of a convolutional neural network (CNN), we have developed an effective approximation method with a Monte Carlo simulation algorithm to tackle the well-known open problem. In doing so, the existence and uniqueness of a 2-tuple adapted strong solution to an approximation B-SPDE were proved. Meanwhile, the convergence of a newly designed simulation algorithm was established. Simulation examples and an application in finance were also provided.

Keywords: backward stochastic partial differential equation (B-SPDE); Monte Carlo simulation, strongly nonlinear; Cauchy terminal value problem; machine learning (ML); convolutional neural network (CNN)
Mathematics Subject Classification: 60H35, 65C30, 60H15, 60K37, 60H30

1. Introduction

In this paper, we study a strongly nonlinear backward stochastic partial differential equation (B-SPDE) in (1.1) through an approximation method and with machine learning (ML)-based Monte Carlo simulation,

$$V(t,x) = H(T,x) - \frac{1}{2}\int_t^T \frac{(V_x(s,x) + \bar{V}_x(s,x))^2}{V_{xx}(s,x)}ds - \int_t^T \bar{V}(s,x)dW(s), \tag{1.1}$$

which has a given terminal random field $H(T, x)$ at time T for $(t, x) \in [0, T] \times R$ and $R = (-\infty, \infty)$ and which is driven by a Brownian motion $W(\cdot)$ over time interval $[0, T]$.

The equation in (1.1) is well-known and was previously derived from studies in finance (see Musiela and Zariphopoulou [22], Øksendal et al. [24], etc.), where, the primary motivation to study the B-SPDE was to solve an optimal portfolio selection problem in an incomplete financial market modeled by a stochastic differential equation driven by multi-dimensional Brownian motion (see, e.g., Musiela and Zariphopoulou [22] and Kramkov and Sirbu [18]). Furthermore, the problem describes the evolution of a related value function and seeks to maximize an expected utility from terminal wealth over admissible strategies. To find an optimal investment policy for this stochastic optimization problem, the well-known dynamic programming principle can be applied to determine such an optimal control policy in a backward way with respect to the time parameter. More precisely, in this financial problem, the random field $V(t, x)$ denotes the value function of an investor's target to maximize his expected utility of terminal wealth over admissible strategy set $\mathcal{A}_T$ (see its definition in Subsection 2.2) with $t \in [0, T]$, i.e.,

$$V(t, x) = \sup_{\beta \in \mathcal{A}_T} E\Big[\mu_T(X^\beta(T))\Big|\mathcal{F}_t, X(t) = x\Big] \tag{1.2}$$

for a given trading horizon $[0, T]$ and the investor's utility $\mu_T(\cdot) : R_+ \to R$ at terminal time T (see more explanations in Subsection 2.2). Note that, in this case, we take $H(T, \cdot) = \mu_T(\cdot)$ in the equation of (1.1). Furthermore, $X^\beta(t)$ denotes the present value of the investor's aggregate investment at time t (see more details in Subsection 2.2). In existing studies (see, e.g., Cerny and Kallsen [2] and Dai [8]), the authors aim to find a so-called variance optimal martingale measure Q^* such that the value function in (1.2) has a simple expression given by the following conditional expectation,

$$V(t, x) = E_{Q^*}\Big[H(T, x)\Big|\mathcal{F}_t, X(t) = x\Big]. \tag{1.3}$$

In this sense, $V(t, x)$ can be decomposed into a macro-trend part and a micro-regulating (volatility) part as shown in (1.1) due to martingale representation theorem (see, e.g., Øksendal [23]). Roughly speaking, the random field $\bar{V}(\cdot, \cdot)$ corresponds to the volatility rate and is the Malliavin derivative of $V(t, x)$ with respect to the Wiener measure corresponding to the Brownian motion W (see more details in Dai [9]).

Due to some difficulties with solving the problem in (1.2) directly and as an alternative to the method presented in (1.3), the authors in Musiela and Zariphopoulou [22] derived the B-SPDE to solve the optimal investment problem. In this sense, the B-SPDE might be considered as the non-Markovian analogue of the traditional Hamilton-Jacobi-Bellman (HJB) equation in Markovian models or in its dual formulation. Once we obtain the solution to the B-SPDE, we can determine the optimal investment policy (see Subsection 2.2 for more details). However, how to analyze and solve Eq (1.1) in general has remained a problem for quite a long time. The main difficulty is due to the singularity of the B-SPDE since it is a strongly nonlinear one. To develop a method to solve this problem, we need to go over the work in Dai [9] and make a comparison between it and our current study.

More precisely, in Dai [9], we developed a generic convolutional neural network (CNN)-based numerical scheme to simulate the 2-tuple adapted strong solution to a unified system of B-SPDEs driven by Brownian motions, which can be applied to many B-SPDE equations. Nevertheless, in

proving the unique existence of the 2-tuple adapted strong solution to the unified system, we need to impose the so-called general local Lipschitz and linear growth conditions. Furthermore, in Dai [9], the generic numerical scheme was developed by a CNN through conditional expectation projection, which is a completely discrete and iterative algorithm in terms of both time and space. However, in estimating the mean-square error and proving the convergence for the CNN-based numerical scheme, we also need the general local Lipschitz and linear growth conditions. Moreover, the generic numerical scheme in Dai [9] does not integrate ML techniques into its computation algorithm.

Although the equation in (1.1) is a special case of our unified system of B-SPDEs in Dai [9], it is a strongly nonlinear one with singularity. Therefore, the B-SPDE in (1.1) does not satisfy the imposed general local Lipschitz and linear growth conditions in Dai [9], which implies that the developed method in Dai [9] cannot be directly applied to solve the equation in (1.1). Hence, by introducing new truncation operators and integrating the ML technique into the CNN platform in Dai [9], we develop an effective approximation method with a Monte Carlo simulation algorithm to tackle such a well-known open problem. Concerning a CNN, readers are also referred to Brizuela and Merchan [12], Dai [10], LeCun et al. [20], Vaswani et al. [27], and Yamashitza et al. [28] for more details. Note that the purpose of integrating the ML technique into our platform is to speed the convergence of our new algorithm designed in the current study (see Algorithm 3.1 and Figure 1 for more details).

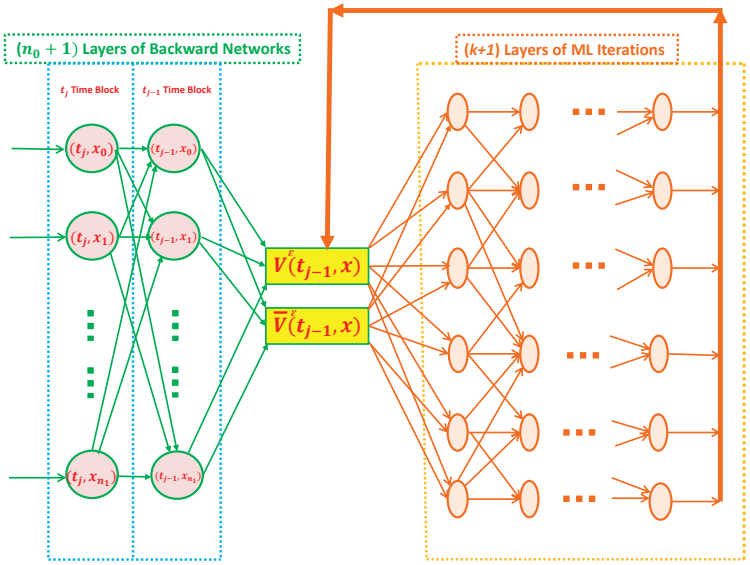

Figure 1. The flow chart of Algorithm 3.1, where the abbreviation "ML" means machine learning.

To go further, we give some explanations about the notations used in the equation of (1.1). The notations V_x, V_{xx}, and $\bar{V}_x$ are the corresponding first-order and second-order partial derivatives of V and $\bar{V}$ in terms of position parameter $x \in R$. Since the second partial derivative V_{xx} appears in the denominator of the second term on the right-hand side of (1.1), the equation in (1.1) is a strongly nonlinear one and exhibits singularity. In addition, since both $V(t, x)$ and $\bar{V}(t, x)$ are unknown random fields, this equation is a diophantine equation (see the related explanation in Dai [9]). Therefore, one major task in studying the equation in (1.1) is to try to obtain an adapted solution pair $(V, \bar{V})$ with

respect to a filtration $\{\mathcal{F}_t, t \geq 0\}$ generated by the Brownian motion, i.e., $\mathcal{F}_t = \sigma(W(s), s \leq t)$. Then, based on this paired solution, we can get its related further optimal financial investment strategy (see the discussions in Musiela and Zariphopoulou [22], Øksendal et al. [24], etc.).

Concerning the relationship between V and $\bar{V}$, we can interpret $\bar{V}$ as a regulating process of V due to the martingale representation theorem (see, e.g., Øksendal [23]). Furthermore, under certain conditions, $\bar{V}$ can be expressed as a functional of the Malliavin derivative of V (see Lemmas 4.6 and 4.7 in Dai [9] for more details). However, based on the Malliavin functional relationship between V and $\bar{V}$, it is difficult to design a direct computation algorithm to calculate $(V, \bar{V})$ numerically due to its complexity. Therefore, in Dai [9], this explicit relationship through Malliavin calculus is only used to prove the convergence of a more directly designed computation algorithm in solving $(V, \bar{V})$ numerically for B-SPDEs under the so-called generalized linear growth and Lipschitz conditions. Nevertheless, in the current study, our B-SPDE in (1.1) is a strongly nonlinear one that does not satisfies the generalized linear growth and Lipschitz conditions. Thus, our main focus in this paper is on the evolution of the study in Dai [9] to solve our current strongly nonlinear problem in terms of algorithm design, analysis, and implementation. The Malliavin functional relationship between V and $\bar{V}$ will not be directly used in this study.

As introduced previously, to the best of our knowledge, the B-SPDE in (1.1) is still not well-solved. Hence, we here try to develop a numerical scheme with related theory to simulate this equation in an approximated way. More precisely, we consider an approximated analog of the equation in (1.1) as follows,

$$V(t,x) = H(T,x) - \frac{1}{2}\int_t^T \frac{(\Phi_{\bar{K}}(V_x(s,x) + \bar{V}_x(s,x)))^2}{\Psi^{\epsilon,K}(V_{xx}(s,x))}ds - \int_t^T \bar{V}(s,x)dW(s) \tag{1.4}$$

for $x \in D = [0,b]$ with $b > 0$, where $\Phi_{\bar{K}}(\cdot)$ is a truncation map corresponding to the first-order derivatives $V_x(t,x)$ and $\bar{V}_x(t,x)$ (if any) for a small number $\epsilon > 0$ and a large number $\bar{K} > 0$. In other words, for $f_x(t,x) = V_x(t,x) + \bar{V}_x(t,x)$, we have that

$$\Phi_{\bar{K}}(f_x(t,x)) \equiv \begin{cases} \bar{K} & \text{if} \quad f_x(t,x) > \bar{K}, \\ f_x(t,x) & \text{if} \quad |f_x(t,x)| \leq \bar{K}, \\ -\bar{K} & \text{if} \quad f_x(t,x) < -\bar{K}, \end{cases} \tag{1.5}$$

where $|\cdot|$ denotes the absolute value of a number $\cdot$. Furthermore, $\Psi^{\epsilon,K}(\cdot)$ is another truncation map corresponding to the second-order derivative $V_{xx}(t,x)$ (if any) for a small number $\epsilon > 0$ and a large number $K > 0$, i.e.,

$$\Psi^{\epsilon,K}(V_{xx}(t,x)) \equiv \begin{cases} V_{xx}(t,x) & \text{if} \quad \epsilon \leq |V_{xx}(t,x)| \leq K, \\ \epsilon & \text{if} \quad 0 \leq V_{xx}(t,x) < \epsilon, \\ -\epsilon & \text{if} \quad -\epsilon < V_{xx}(t,x) < 0, \\ K & \text{if} \quad V_{xx}(t,x) > K, \\ -K & \text{if} \quad V_{xx}(t,x) < -K. \end{cases} \tag{1.6}$$

From (1.6), we can see that the absolute value of the truncation map $\Psi^{\epsilon,K}(\cdot)$ is always greater or equal to the positive number ϵ (i.e., $|\Psi^{\epsilon,K}(\cdot)| \geq \epsilon$). Thus, the denominator appearing in (1.4) is always away from zero, which implies that the equation for the given constants $\epsilon > 0$, $K > 0$, and $\bar{K} > 0$ has the

potential to be well-behaved. From (1.5) and (1.6), we can see that the truncation maps $\Phi_{\bar{K}}(\cdot)$ and $\Psi^{\epsilon,K}(\cdot)$ are both bounded for the given constants $\epsilon > 0$, $K > 0$, and $\bar{K} > 0$. Furthermore, based on the truncated maps in (1.5) and (1.6) and the mentioned properties, we can prove that the equation in (1.4) satisfies the generalized local linear growth and generalized local Lipschitz conditions as given in Dai [9]. Hence, under a suitable terminal condition, there uniquely exists a $\{\mathcal{F}_t\}$-adapted strong solution $(V^{\epsilon,K}_{\bar{K}}(t,x), \bar{V}^{\epsilon,K}_{\bar{K}}(t,x))$ to the equation in (1.4), which implies that this equation in (1.4) is not a singular one for the given constants $\epsilon > 0$, $K > 0$, and $\bar{K} > 0$. Thus, according to the equation in (1.4), we can develop a Monte Carlo simulation algorithm by enhancing the one developed in Dai [9] through adding an additional machine learning (ML) loop (see Algorithm 3.1 in Section 3 for more details). Note that the design of using a sequence of structure-preserving B-SPDEs in (1.4) to approximate the one in (1.1) is actually motivated from the diffusion approximation for queueing networks where r is used to index a specific network. More exactly, the diffusion approximation is to approximate a target limit system through a sequence of physical systems when r tends to infinity (see, e.g., Dai [4, 5, 7]). Furthermore, during the approximation, each system keeps its structure unchanged.

In summary, the contributions of our paper can be stated in two folds: theoretic contributions and numerical contributions.

Concerning the theoretic contributions of our paper, we study the strongly nonlinear B-SPDE in (1.1) through an approximation method by introducing new truncation operators as in (1.5) and (1.6). The main reason for doing it in this way is due to the singularity of the B-SPDE in (1.1). To guarantee the meaningfulness of the approximation, the existence and uniqueness of a 2-tuple adapted strong solution to the approximation B-SPDE in (1.4) are proved. Furthermore, it is also proved that, if the B-SPDE in (1.1) has a solution pair $(V, \bar{V})$, the solution pair to the approximation B-SPDE in (1.4) will converge to $(V, \bar{V})$ as ϵ tends to zero and $K, \bar{K}$ tend to the infinity in certain senses. In addition, based on the approximated B-SPDE in (1.4), some investment policies are analytically derived for a financial market with the aim to conduct some accuracy comparisons.

Concerning the numerical contributions of our paper, we develop a Monte Carlo simulation-based algorithm to numerically solve the approximation B-SPDE in (1.4). In doing so, we integrate an additional ML loop into the platform of a convolutional neural network (CNN) as studied in Dai [9]. Based on the equation in (1.4), we can prove the convergence of this newly designed algorithm in the case that the equation in (1.1) has an adapted strong solution. The basic idea to develop such an algorithm is to find a CNN denoted by $\mathcal{U}$ with an additional ML loop at each time point to approximate $(V^{\epsilon,K}_{\bar{K}}, \bar{V}^{\epsilon,K}_{\bar{K}})$ (see Figure 1 with explanations in Subsection 3.1 for more details). In this CNN, we consider $(V^{\epsilon,K}_{\bar{K}}, \bar{V}^{\epsilon,K}_{\bar{K}})$ as paired parameters to be trained at each node (t,x) (also called neuron (t,x)). The learning strategy for this CNN is to minimize the mean-squared error (loss) between $(V^{\epsilon,K}_{\bar{K}}, \bar{V}^{\epsilon,K}_{\bar{K}})$ and $\mathcal{U}$. Note that, in Dai [9], the CNN corresponding to $\mathcal{U}$ is expressed as a conditional expectation projection for a given data set. To compute the conditional expectation, the well-known tower law (see, e.g., Kallenberg [16]) is used to design the CNN $\mathcal{U}$ (see Subsection 3.1 for more details), which is a backward one. Similar to a multi-layer perceptron (MLP) as studied in Cybenko [3], Haykin [13], and Hornik et al. [14], our CNN $\mathcal{U}$ is a fully connected neural network. However, it is still computationally efficient since our B-SPDE in (1.4) is a Browinian motion driven one. Therefore, the computation of our conditional expectation does not need the additional multiple parameter training since the transition probability and density of a Brownian motion is given. As an alternative, our conditional expectation projection can also be presented by a linear or nonlinear regression model with parameters, which

is closely related to the so-called Kolmogorov-Arnold network due to Kolmogorov's superposition theorem (see, e.g., Kolmogorov [17] and Braun and Griebel [1]) and the recent study in Liu et al. [21]. If we do in this way, we need to train additional parameters, which is time-consuming and may sacrifice some accuracy. Thus, in Dai [9] and in our current paper, we report our achievements based on our successfully implemented and tower-law-oriented CNN $\mathcal{U}$. Furthermore, along this line, we refer the reader to these related papers Gonon et al. [11], Kratsios et al. [19], Peluchetti [25], Sirignano and Spiliopoulos [26], Vaswani et al. [27] for more details. Finally, concerning the numerical contributions of this study, we also provide simulation examples supported with an application in finance.

For the reader's convenience, the partial simulation results according to our developed simulation scheme are shown in Figure 2, where we present the simulation results concerning the paired solution $(V_{\bar{K}}^{\epsilon,K}, \bar{V}_{\bar{K}}^{\epsilon,K})$ to the approximated B-SPDE in (1.4) with $\epsilon = 1/100000$, $K = 10^9$, and $\bar{K} = 2^{64}$. The computed values at time point $t_{j_0} = t_{n_0}$ with $n_0 = 10$ correspond to the input terminal values at $t_{n_0+1} = T$. The "solution error check" titled in the third plot of the first column in Figure 2 is in terms of the difference between the two sides of the B-SPDE in (1.4) and is with respect to a particular sample path. From the simulation results displayed in the third plot of the first column, we can see that our algorithm is quite accurate. The three plots in the first row of Figure 2 display the simulated $V_{\bar{K}}^{\epsilon,K}(t_{n_0}, x)$ together with its simulated first-order and second-order derivatives in terms of position parameter $x \in \{b/d, 2b/d..., (d - dropnum)b/d\}$ with $b = 1/6000$, $d = 100$, and $dropnum = 30$. Although the graph in the third plot is non-smooth, it is close to a smooth line. Furthermore, the three plots in the second row of Figure 2 display the simulated $\bar{V}_{\bar{K}}^{\epsilon,K}(t_{n_0}, x)$ together with its simulated first-order and second-order derivatives. In addition, the second and third plots in the third row of Figure 2 display the simulated terminal value $V_{\bar{K}}^{\epsilon,K}(t_{n_0+1}, x)$ together with its simulated first-order derivative. Concerning the simulations about the paired solution $(V_{\bar{K}}^{\epsilon,K}(t, x), \bar{V}_{\bar{K}}^{\epsilon,K}(t, x))$, readers are referred to Section 4 for more details.

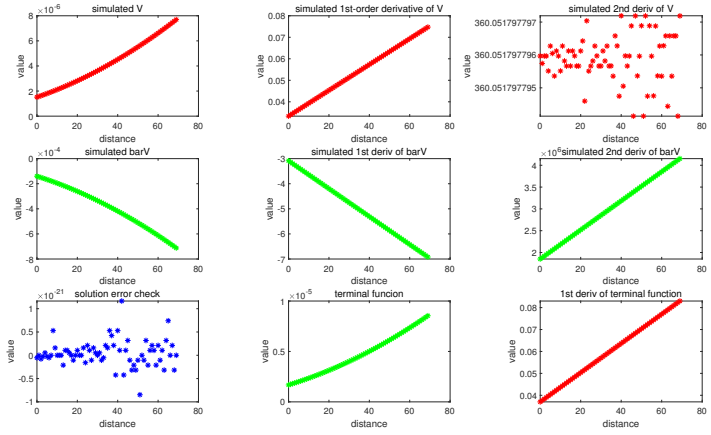

Figure 2. Simulated solution pair and pathwise error comparison to the B-SPDE in (1.4) with $\epsilon = 1/100000$, $K = upperbound = 10^9$, $\bar{K} = 2^{64}$, and $learningrate = 1/2$. Furthermore, $T = 0.1$, $n = 60000$, $hhh = 2$, $terminalcoefficient = 2 * 100$, $Q = 2352$, $b = 1/6000$, $d = 100$, $dropnum = 30$, $BMD = 2$, $bmdp = 3000$, and $k = 1$. In this figure, we display the evolution results of $(d - dropnum)$ points with respect to the position parameter $x \in \{b/d, 2b/d, ..., (d - dropnum)b/d\}$ over horizon-axes at a particular time point $T(n - 1)/n$.

As pointed out previously, the primary interest in deriving the B-SPDE in (1.1) is to find the financial investment strategy together with myopic investment and excess hedging demand (see, e.g., Dai [6, 8], Musiela and Zariphopoulou [22], and Øksendal et al. [24]). Therefore, based on the simulation study conducted in this paper and the related strategy study presented in Musiela and Zariphopoulou [22], we can also obtain the simulated financial control strategies as presented in Figure 3. In this figure, we display the simulated investment policy, myopic policy, and excess hedging demand at time point t_{j_0} with $j_0 = n_0$ and $n_0 = 10$. These policies correspond to the formula derived in Musiela and Zariphopoulou [22]. The three graphs in the left column are corresponding to the simulated pathwise results. The three graphs in the right column correspond to the simulated results in the mean average sense with respect to the simulation iteration number Q. Theoretically, the myopic policy should continue to be constant. Our simulated results support this theoretical result. However, this theoretic result also further justifies the correctness of our algorithm and simulations. Concerning this financial policy simulation, readers are referred to Section 4 for more details.

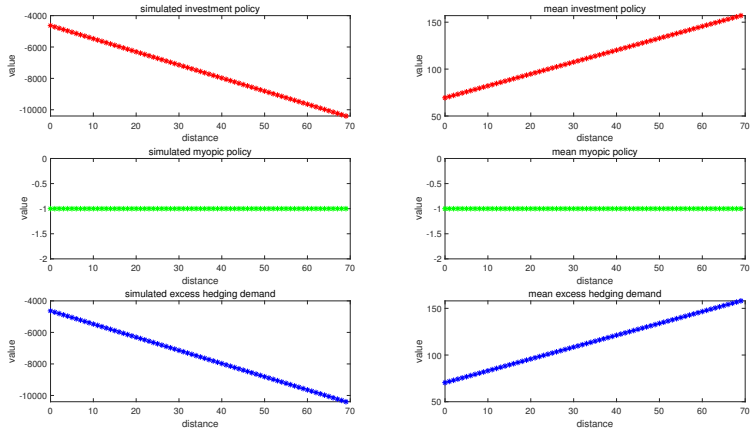

Figure 3. Simulated investment policy, myopic policy, and excess hedging demand with $\epsilon = 1/100000$, $K = upperbound = 10^9$, $\bar{K} = 2^{64}$, and $learningrate = 1/2$. Furthermore, $T = 0.1$, $n = 60000$, $hhh = 2$, $terminalcoefficient = 2 * 100$, $Q = 2352$, $b = 1/6000$, $d = 100$, $dropnum = 30$, $BMD = 2$, $bmdp = 3000$, and $k = 1$. In this figure, we display the evolution results of $(d - dropnum)$ points with respect to the position parameter $x \in \{b/d, 2b/d, ..., (d - dropnum)b/d\}$ over horizon-axes at a particular time point $T(n-1)/n$.

The remainder of the paper is organized as follows. In Section 2, we present the unique existence theorem with proof of our approximated B-SPDE. Furthermore, in this section, we also present the application of our study in finance. In Section 3, we design our Monte Carlo simulation algorithm through CNN and ML. The convergence of our designed simulation algorithm is also proved. In Section 4, we present our numerical simulation case studies. Finally, in Section 5, we summarize our study conducted in this paper with conclusions.

2. Unique existence theorem with an application in finance

This section consists of two subsections. The unique existence theorem is stated in Subsection 2.1. The financial application of our unique existence result is presented in Subsection 2.2.

2.1. Unique existence theorem

Let $C^2(D,R)$ be the Banach space of all functions f having continuous derivatives up to the order 2 with the uniform norm,

$$\|f\|_{C^2(D,R)} = \max_{c\in\{0,1,2\}} \max_{j\in\{1,\ldots,r(c)\}} \sup_{x\in D} \left|f_j^{(c)}(x)\right| \tag{2.1}$$

for each $f \in C^2(D,R)$. The $r(c)$ in (2.1) for each $c \in \{0,1,2\}$ is the total number of the partial derivatives of the cth order. Then, we can introduce our required measurable spaces used in this paper. First, we use $L^2_{\mathcal{F}}([0,T],C^2(D;R))$ to denote the set of all R-valued (also called $C^2(D;R)$-valued) measurable random fields $Z(t,x)$ satisfying

$$E\left[\int_0^T \|Z(t)\|^2_{C^2(D,R)}dt\right] < \infty, \tag{2.2}$$

where the random field $Z(t,x)$ is assumed to be adapted to $\{\mathcal{F}_t, t\in[0,T]\}$ for each $x \in D$ and $Z(t,x) \in C^2(D,R)$ with a fixed $t \in [0,T]$). Second, we use $L^2_{\mathcal{F},1}([0,T],C^2(D,R))$ to denote the corresponding set of predictable processes (see the definitions on pages 21 and 45 of Ikeda and Watanabe [15]). Third, we use $L^2_{\mathcal{F}_T}(\Omega,C^2(D;R))$ to denote the set of all R-valued, $\mathcal{F}_T$-measurable random fields $\zeta(x,\omega)$ for each $x \in D$ and sample point $\omega \in \Omega$, where $\zeta(x,\omega) \in C^2(D,R)$ for each $\omega \in \Omega$ satisfies

$$\|\zeta\|^2_{L^2_{\mathcal{F}_T}(\Omega,C^2(D,R))} \equiv E\left[\|\zeta\|^2_{C^2(D,R)}\right] < \infty. \tag{2.3}$$

Therefore, we can introduce our supporting space as follows,

$$\bar{\mathcal{Q}}^2_{\mathcal{F}}([0,T]\times D) \equiv L^2_{\mathcal{F}}([0,T],C^2(D,R)) \times L^2_{\mathcal{F},1}([0,T],C^2(D,R)). \tag{2.4}$$

Finally, before introducing our unique existence theorem, we suppose that the terminal value in (1.1) (and hence in (1.4)) is given by

$$H(x) = h_1(x)h_2(W(T)), \tag{2.5}$$

where both h_1 and h_2 are polynomials (interested readers are also referred to Dai [9] for more related explanation). Then, we can state our unique existence theorem as follows.

Theorem 2.1. *Under the terminal condition in (2.5), there is a unique strong $\{\mathcal{F}_t\}$-adapted solution pair $(V^{\epsilon,K}_{\bar{K}}(t,x), \bar{V}^{\epsilon,K}_{\bar{K}}(t,x))$ to the B-SPDE in (1.4) within the space $\bar{\mathcal{Q}}^2_{\mathcal{F}}([0,T]\times D)$ for a small constant $\epsilon > 0$, a large number $K > 0$, and a large number $\bar{K} > 0$ with $t \in [0,T]$. Furthermore, we have that*

$$\sup_{t\in[0,T]} E\left[\left\|V^{\epsilon,K}_{\bar{K}}(t)\right\|^2_{C^2(D,R)}\right] < \infty. \tag{2.6}$$

Proof. Corresponding to the B-SPDE in (1.4), we define a second-order partial differential operator $\mathcal{L}^{\epsilon,K}_{\bar{K}}$ as follows,

$$\mathcal{L}^{\epsilon,K}_{\bar{K}}(t,x,V,\bar{V}) = \frac{(\Phi_{\bar{K}}(V_x(t,x)+\bar{V}_x(t,x)))^2}{\Psi^{\epsilon,K}(V_{xx}(t,x))}. \tag{2.7}$$

Then, we can conclude that the operator $\mathcal{L}_{\bar{K}}^{\epsilon,K}$ defined in (2.7) satisfies the generalized local Lipschitz and linear growth conditions as introduced in Dai [9], i.e.,

$$\begin{aligned} \left|\Delta\mathcal{L}_{\bar{K}}^{\epsilon,K}(s,x,(u,\bar{u}),(v,\bar{v}))\right| &\leq K_D\left(\|u-v\|_{C^2(D,R)}+\|\bar{u}-\bar{v}\|_{C^2(D,R)}\right), \quad (2.8)\\ \left|\mathcal{L}_{\bar{K}}^{\epsilon,K}(s,x,(u,\bar{u}))\right| &\leq K_D\left(\|u\|_{C^2(D,R)}+\|\bar{u}\|_{C^2(D,R)}\right) \quad (2.9)\end{aligned}$$

for each fixed $(t,x)\in[0,T]\times D$, and any $(u,\bar{u})$, $(v,\bar{v})\in C^2(D,R)\times C^2(D,R)$, where $(u,\bar{u})$ and $(v,\bar{v})$ are two pairs corresponding to the equation in (1.4). Furthermore, K_D is a nonnegative constant depending on D, ϵ, K, and $\bar{K}$. In addition, the operator $\Delta\mathcal{L}_{\bar{K}}^{\epsilon,K}$ is defined by

$$\Delta\mathcal{L}_{\bar{K}}^{\epsilon,K}(s,x,(u,\bar{u}),(v,\bar{v}))\equiv\mathcal{L}_{\bar{K}}^{\epsilon,K}(s,x,(u,\bar{u}))-\mathcal{L}_{\bar{K}}^{\epsilon,K}(s,x,(v,\bar{v})) \quad (2.10)$$

for each given $(t,x,(u,\bar{u}),(v,\bar{v})))$.

In fact, for any two pairs of $(U,\bar{U})$ and $(V,\bar{V})$ in the space $\bar{\mathcal{Q}}_{\mathcal{F}}^2([0,T]\times D)$, it follows from the definition in (2.7) that we can prove the claim in (2.8) as follows,

$$\begin{aligned} &\left|\Delta\mathcal{L}_{\bar{K}}^{\epsilon,K}(s,x,(U,\bar{U}),(V,\bar{V}))\right|\\ &\leq\left|\frac{\left(\Phi_{\bar{K}}(U_x(s,x)+\bar{U}_x(s,x))\right)^2}{\Psi^{\epsilon,K}(U_{xx}(s,x))}-\frac{\left(\Phi_{\bar{K}}(V_x(s,x)+\bar{V}_x(s,x))\right)^2}{\Psi^{\epsilon,K}(V_{xx}(s,x))}\right|\\ &=\left|\frac{\Psi^{\epsilon,K}(V_{xx}(s,x))\left(\Phi_{\bar{K}}(U_x(s,x)+\bar{U}_x(s,x))\right)^2}{\Psi^{\epsilon,K}(U_{xx}(s,x))\Psi^{\epsilon,K}(V_{xx}(s,x))}\right.\\ &\left.-\frac{\Psi^{\epsilon,K}(U_{xx}(s,x))\left(\Phi_{\bar{K}}(V_x(s,x)+\bar{V}_x(s,x))\right)^2}{\Psi^{\epsilon,K}(U_{xx}(s,x))\Psi^{\epsilon,K}(V_{xx}(s,x))}\right|\\ &\leq\frac{1}{\epsilon^2}\left|\Psi^{\epsilon,K}(V_{xx}(s,x))\left(\left(\Phi_{\bar{K}}(U_x(s,x)+\bar{U}_x(s,x))\right)^2-\left(\Phi_{\bar{K}}(V_x(s,x)+\bar{V}_x(s,x))\right)^2\right)\right| \quad (2.11)\\ &+\frac{1}{\epsilon^2}\left|\left(\Psi^{\epsilon,K}(U_{xx}(s,x))-\Psi^{\epsilon,K}(V_{xx}(s,x))\right)\left(\Phi_{\bar{K}}(V_x(s,x)+\bar{V}_x(s,x))\right)^2\right|\\ &\leq K_D^1\left(\left|\Psi^{\epsilon,K}(U_{xx}(s,x))-\Psi^{\epsilon,K}(V_{xx}(s,x))\right|\right.\\ &+\left.\left|\Phi_{\bar{K}}(U_x(s,x)+\bar{U}_x(s,x))-\Phi_{\bar{K}}(V_x(s,x)+\bar{V}_x(s,x))\right|\right)\\ &\leq K_D\left(\|u-v\|_{C^2(D,R)}+\|\bar{u}-\bar{v}\|_{C^2(D,R)}\right),\end{aligned}$$

where K_D^1 is some positive constant depending on ϵ, K, $\bar{K}$, and D. Furthermore, K_D is a positive constant depending on K_D^1.

Similarly, by applying the definition in (2.7), we can prove the claim in (2.9) as follows,

$$\left|\mathcal{L}_{\bar{K}}^{\epsilon}(s,x,(U,\bar{U}))\right|\leq\frac{1}{\epsilon}\left(\Phi_{\bar{K}}(U_x(t,x)+\bar{U}_x(t,x))\right)^2\leq K_D\left(\|u\|_{C^2(D,R)}+\|\bar{u}\|_{C^2(D,R)}\right),$$

where the constant K_D in (2.11) and (2.12) can be chosen such that the inequalities in (2.11) and (2.12) are both true.

Finally, it follows from (2.11), (2.12), and the discussion in Dai [9] that the claims in our main theorem are true. Hence, we complete the proof of Theorem 2.1. □

2.2. Application in finance

Consider a financial market consisting of one risky asset and one riskless asset. The risky asset is a stock whose price dynamics is driven by the Brownian motion $W(\cdot)$, i.e.,

$$dS(t) = S(t)\Big(u(t)dt + \sigma(t)dW(t)\Big) \tag{2.12}$$

with initial price $S(0) > 0$. Furthermore, $u(\cdot)$ and $\sigma(\cdot)$ are $\{\mathcal{F}_t\}$-progressive measurable processes with values in $R = (-\infty, \infty)$. The riskless asset is with the price process $R(t)$ with an interest rate $r(t)$, i.e.,

$$dR(t) = r(t)R(t)dt. \tag{2.13}$$

Then, it follows from the discussion in Musiela and Zariphopoulou [22] that the present value $X^{\beta}(t) = \beta^0(t) + \beta^1(t)$ of the aggregate investment concerning the riskless investment strategy $\beta^0(t)$ and the risky investment strategy $\beta^1(t)$ is given by

$$dX^{\beta}(t) = \beta(t)(u(t) - r(t))dt + \sigma(t)\beta(t)dW(t), \tag{2.14}$$

where $\beta(t) = \beta^1(t)$ is the discounted strategy in the following admissibility set,

$$\begin{aligned} \mathcal{A} \;=\; \Big\{&\beta : \beta(t) \text{ is self-financing and } \{\mathcal{F}_t\}\text{-progressively measurable} \\ &\text{satisfying } E\left[\int_0^t |\sigma(s)\beta(s)|^2\right] < \infty,\ X^{\beta}(t) \geq 0,\ t \geq 0\Big\}. \end{aligned} \tag{2.15}$$

Now, for a given trading horizon $[0, T]$ and an investor's utility $\mu_T(\cdot) : R_+ \to R$ at terminal time T, which is supposed to be an increasing and convex function of his wealth, we can represent the risk-seeking attitude of an investor (the risk-avoiding case corresponding to a concave function can be similarly discussed, see, e.g., Musiela and Zariphopoulou [22] for a reference). The investor's target is to maximize the expected utility of terminal wealth over the admissible strategy set $\mathcal{A}_T$ corresponding to the one in (2.15) with $t \in [0, T]$, i.e., to solve the optimization problem presented in (1.2). Then, if we take $u(t) = \sigma(t) \equiv 1$ and $r(t) \equiv 0$, respectively, in (2.12) and (2.13), the optimal equation corresponding to the optimization problem in (1.2) is given by the B-SPDE in (1.1) with $H(T, x) = \mu_T(x)$ (concerning the justification of this claim, readers are referred to Musiela and Zariphopoulou [22] for more details). In this case, the optimal feedback investment strategy is given by

$$\beta^*(t) = -\frac{V_x(t, x)}{V_{xx}(t, x)} - \frac{\bar{V}_x(t, x)}{V_{xx}(t, x)}, \tag{2.16}$$

where the first term (will be denoted by $\beta^{*,m}(t)$) on the right-hand-side of (2.16) is called a myopic investment strategy resembling the investment policy followed by an investor in a financial market where the investment opportunity set keeps constant through time. Furthermore, the second term (will be denoted by $\beta^{*,h}(t)$) on the right-hand-side of (2.16) is referred as the excess hedging demand denoting the additional investment due to the volatility $\bar{V}(t, x)$ of the performance process $V(t, x)$. Thus,

corresponding to the paired solution $(V_{\bar{K}}^{\epsilon,K}(t,x), \bar{V}_{\bar{K}}^{\epsilon,K}(t,x))$ to the equation in (1.4), the approximated optimal investment strategies can be denoted by $\beta_{\epsilon}^{*}(t)$, $\beta_{\epsilon}^{*,m}(t)$, and $\beta_{\epsilon}^{*,h}(t)$ (see, e.g., the simulation results in Figures 3–5).

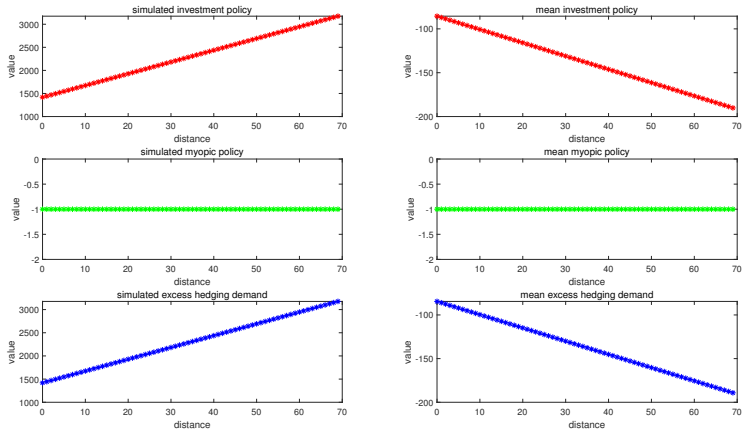

Figure 4. Simulated investment policy, myopic policy, and excess hedging demand with $\epsilon = 1/100000$, $K = upperbound = 10^9$, $\bar{K} = 2^{64}$, and $learningrate = 1/2$. Furthermore, $T = 0.1$, $n = 60000$, $hhh = 2$, $terminalcoefficient = 2 * 100$, $Q = 2352$, $b = 1/6000$, $d = 100$, $dropnum = 30$, $BMD = 2$, $bmdp = 3000$, and $k = 1$. In this figure, we display the evolution results of $(d - dropnum)$ points with respect to the position parameter $x \in \{b/d, 2b/d, ..., (d - dropnum)b/d\}$ over horizon-axes at a particular time point $T(n-8)/n$.

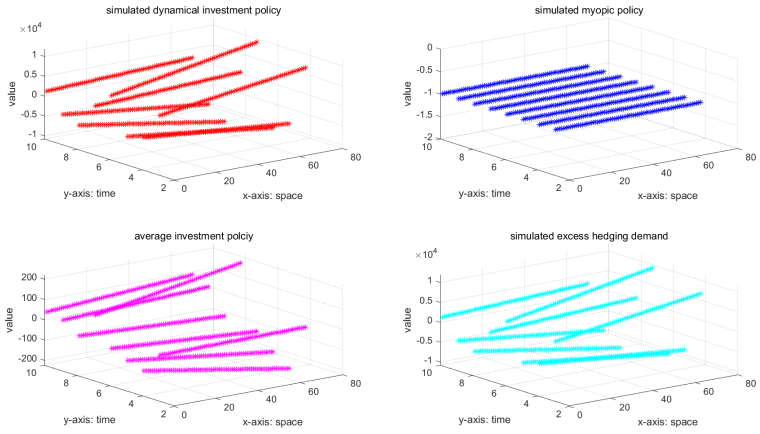

Figure 5. Simulated dynamical evolutions of investment policy, myopic policy, and excess hedging demand with $\epsilon = 1/100000$, $K = upperbound = 10^9$, $\bar{K} = 2^{64}$, and $learningrate = 1/2$. Furthermore, $T = 0.1$, $n = 60000$, $hhh = 2$, $terminalcoefficient = 2 * 100$, $Q = 2352$, $b = 1/6000$, $d = 100$, $dropnum = 30$, $BMD = 2$, $bmdp = 3000$, and $k = 1$. In this figure, we display the evolution results of $(d - dropnum)$ points with respect to the position parameter $x \in \{b/d, 2b/d, ..., (d - dropnum)b/d\}$ over horizon-axes and time parameter $t \in \{T, T(n-1)/n, ..., T(n-8)/n\}$.

3. The Monte Carlo simulation algorithm and its convergence

This section consists of two subsections concerning the design of a Monte Carlo simulation algorithm and proving its convergence with error bound estimation.

3.1. Simulation algorithm

In this subsection, we develop a Monte Carlo simulation algorithm based on both CNN and machine learning (see Figure 1) to simulate the 2-tuple adapted strong solution to the B-SPDE in (1.4). More precisely, we consider a partition π for the product region of $[0,T]\times D$ with $D=[0,b]$ as follows,

$$\begin{aligned}\pi: \quad & 0=t_0<t_1<\cdots<t_{n_0}=T \quad \text{with } n_0\in\{0,1,...\},\\ & 0=x_0<x_1<\cdots<x_{n_1}=b \quad \text{with } n_1\in\{0,1,...\},\end{aligned} \tag{3.1}$$

where t_{j_0} for $j_0\in\{0,1,2,...,n_0\}$ and x_{j_1} for $j_1\in\{0,1,...,n_1\}$ are the points of divisions over the time interval $[0,T]$ and the space interval D. Then, for all $j_l\in\{1,...,n_l\}$ with $l\in\{0,1\}$, we take

$$\Delta_{j_0}^{t,\pi} = t_{j_0}-t_{j_0-1}, \tag{3.2}$$

$$\Delta_1^{\pi} = x_{j_1}-x_{j_1-1}=\frac{b}{n_1}, \tag{3.3}$$

$$\Delta^{\pi}W_{j_0} = W(t_{j_0})-W(t_{j_0-1}). \tag{3.4}$$

Furthermore, let

$$|\pi| \equiv \max_{j_0\in\{1,...,n_0\}}\left\{\Delta_{j_0}^{t,\pi},\ \Delta_1^{\pi}\right\}, \tag{3.5}$$

$$D^{j_1} \equiv \left[x_{j_1-1},\ x_{j_1}\right), \tag{3.6}$$

$$\mathcal{X} \equiv \left\{x_{j_1}:\ j_1\in\{0,1,...,n_1\}\right\}. \tag{3.7}$$

Now, we use the forward and the backward difference techniques to approximate the partial derivatives appearing in (1.4). More precisely, for each $f\in\{V_{\bar{K}}^{\epsilon,K},\bar{V}_{\bar{K}}^{\epsilon,K}\}$, $x\in\mathcal{X}$, and each integer $c\in\{1,2\}$, we define the c^{th}-quotient of differences, which corresponds to the c^{th}-order derivative of f along the x direction, as follows,

$$f_\pi^{(c)}(t,x) \equiv \begin{cases} \frac{f_\pi^{(c-1)}(t,x+\Delta_1^\pi)-f_\pi^{(c-1)}(t,x)}{\Delta_1^\pi} & \text{if } x=x_{j_1} \text{ and } j_1<n_1,\\ -\frac{f_\pi^{(c-1)}(t,x-\Delta_1^\pi)-f_\pi^{(c-1)}(t,x)}{\Delta_1^\pi} & \text{if } x=x_{j_1} \text{ and } j_1=n_1,\end{cases} \tag{3.8}$$

where we adopt the convention that $f_\pi^{(0)}=f_\pi$. Furthermore, to simplify the notations, we use $\{V^\epsilon,\bar{V}^\epsilon\}$ to denote $\{V_{\bar{K}}^{\epsilon,K},\bar{V}_{\bar{K}}^{\epsilon,K}\}$, and we define

$$\mathcal{L}_{\bar{K}}^{\epsilon,K}\left(t,x,V_\pi^\epsilon(t,x)\right)\equiv\mathcal{L}_{\bar{K}}^{\epsilon,K}(t,x,(V_\pi^\epsilon(t,x),V_\pi^{\epsilon,(1)}(t,x),V_\pi^{\epsilon,(2)}(t,x)),(\bar{V}_\pi^\epsilon(t,x),\bar{V}_\pi^{\epsilon,(1)}(t,x))) \tag{3.9}$$

for each $x\in\mathcal{X}$. Moreover, we use $\mathcal{L}_{\pi,\bar{K}}^{\epsilon,K}$ to denote the fully discretized version of $\mathcal{L}_{\bar{K}}^{\epsilon,K}$. Then, we can present the following Monte Carlo simulation algorithm.

Algorithm 3.1. *This algorithm consists of three parts:* **Part I**, **Part II**, *and* **Part III**:
Part I. *This part is an iterative one in terms of* $\{(V^{\epsilon}(t_{j_0}, x), \bar{V}^{\epsilon}(t_{j_0}, x)): x \in \mathcal{X}\}$ *with* j_0 *decreasing from* n_0 *to 1 in a backward way,*

$$V_{\pi}^{\epsilon}(t_{n_0}, x) = H_{\pi}(x), \tag{3.10}$$

$$\bar{V}_{\pi}^{\epsilon}(t_{n_0}, x) = 0, \tag{3.11}$$

$$V_{\pi}^{\epsilon}(t_{j_0-1}, x) = E\left[V_{\pi}^{\epsilon}(t_{j_0}, x) + \mathcal{L}_{\pi,\bar{K}}^{\epsilon,K}(t_{j_0}, x, V_{\pi}^{\epsilon}(t_{j_0}, x))\Delta_{j_0}^{t,\pi} \middle| \mathcal{F}_{t_{j_0-1}} \right], \tag{3.12}$$

$$\begin{aligned} \bar{V}_{\pi}^{\epsilon}(t_{j_0-1}, x) = & \frac{1}{\Delta_{j_0}^{\pi}} E\left[V_{\pi}^{\epsilon}(t_{j_0}, x)\Delta^{\pi} W_{j_0} \middle| \mathcal{F}_{t_{j_0-1}} \right] \\ & + E\left[\mathcal{L}_{\pi,\bar{K}}^{\epsilon,K}(t_{j_0}, x, V_{\pi}^{\epsilon}(t_{j_0}, x))\Delta^{\pi} W_{j_0} \middle| \mathcal{F}_{t_{j_0-1}} \right]. \end{aligned} \tag{3.13}$$

Part II. *This part is a machine learning loop at time* t_{j_0-1} *in order to minimize the difference concerning the values on both sides of the equation in (1.4) along each sample path, i.e.,*

$$V_{\pi}^{\epsilon,k+1}(t_{j_0-1}, x) = V_{\pi}^{\epsilon,k}(t_{j_0-1}, x) - \alpha \nabla G(V_{\pi}^{\epsilon,k}(t_{j_0-1}, x)) \tag{3.14}$$

for each $k \in \{0, 1, 2, ...\}$ *with* $V_{\pi}^{\epsilon,0}(t_{j_0-1}, x) = V_{\pi}^{\epsilon}(t_{j_0-1}, x)$*, where* α *is a given learning rate and* $\nabla G(\cdot)$ *is the stochastic gradient of an optimization problem, i.e.,*

$$\min_{V_{\pi}^{\epsilon,k}(t_{j_0-1},x)\in R} G(V_{\pi}^{\epsilon,k}(t_{j_0-1}, x))$$

with its objective function $G(V_{\pi}^{\epsilon,k}(t_{j_0-1}, x))$ *given by*

$$\left(V_{\pi}^{\epsilon,k}(t_{j_0-1}, x) - \left(V_{\pi}^{\epsilon}(t_{j_0}, x) - \frac{1}{2}\mathcal{L}_{\pi,\bar{K}}^{\epsilon,K}(t_{j_0-1}, x, V_{\pi}^{\epsilon,k}(t_{j_0-1}, x))\Delta_{j_0}^{t,\pi} \right) + \bar{V}_{\pi}^{\epsilon}(t_{j_0-1}, x)\Delta^{\pi} W_{j_0} \right)^2.$$

The machine learning loop in this part has a stopping rule as follows.

$$\begin{aligned} & \textit{Choose a number } (k+1) \textit{ to stop the iteration,} \\ & \textit{then take the new } V_{\pi}^{\epsilon}(t_{j_0-1}, x) \textit{ to be } V_{\pi}^{\epsilon,k+1}(t_{j_0-1}, x). \end{aligned} \tag{3.15}$$

Part III. *This part is to compute the numerical derivatives at each time* t_{j_0-1}*, i.e.,*

$$\begin{aligned} & \textit{Compute } V_{\pi}^{\epsilon,(c)}(t_{j_0-1}, x) \textit{ and } \bar{V}_{\pi}^{\epsilon,(c)}(t_{j_0-1}, x) \\ & \textit{for each } x \in \mathcal{X} \textit{ with } c \in \{1, 2\} \textit{ via the formula in (3.8).} \end{aligned} \tag{3.16}$$

Concerning the architecture, the loss, and the learning strategy corresponding to Algorithm 3.1, the associated flow chart is shown in Figure 1. In this flow chart, we present a backward CNN supported with a ML loop.

As displayed in the graph on the left-hand side of Figure 1, this CNN denoted by $\mathcal{U}$ has $(n_0 + 1)$-layers arranged in a backward way, which corresponds to Part I of Algorithm 3.1. In this CNN, a node (also called a neuron) is indexed by (t_j, x_i) with $j \in \{n_0, n_0 - 1, ..., 1, 0\}$ and $i \in \{0, 1, ..., n_1\}$, where, in our Algorithm 3.1, we take $j = j_0$ and $i = j_1$. Associated with each node (t_j, x_i), the paired solution

$(V^\epsilon(t_j, x_i), \bar{V}^\epsilon(t_j, x_i))$ is considered as a pair of parameters to be trained or estimated. The training process is arranged in a backward way as follows,

$$(V^\epsilon(t_{n_0}, x_i), \bar{V}^\epsilon(t_{n_0}, x_i)) \to (V^\epsilon(t_{n_0-1}, x_i), \bar{V}^\epsilon(t_{n_0-1}, x_i)) \to \cdots \to (V^\epsilon(t_0, x_i), \bar{V}^\epsilon(t_0, x_i))$$

for each $i \in \{0, 1, ..., n_1\}$. The design rational for this CNN is the tower law for expectation (see, e.g., Theorem 5.1 (vii) on page 81 of Kallenberg [16]) and more explanations concerning the design rational for a CNN can be found in Dai [9]. The learning strategy for this CNN is to minimize the mean-squared error (loss) between $(V^\epsilon, \bar{V}^\epsilon)$ and $\mathcal{U}$.

As displayed in the graph on the right-hand side of Figure 1, the supporting ML loop corresponds to Part II of Algorithm 3.1. The purpose of adding this ML loop into Algorithm 3.1 is to speed up the convergence of the algorithm in Dai [9]. Actually, in Dai [9], the algorithm is designed only through the backward CNN corresponding to Part I of Algorithm 3.1. Interestingly, our numerical implementations presented in this paper indicate that the convergence of Algorithm 3.1 is indeed faster after adding this ML loop. Concerning the supporting ML loop, the iteration is only designed for V_π^ϵ as shown in Part II of Algorithm 3.1. In this iteration, we keep $\bar{V}_\pi^\epsilon$ the same for all k. In this ML loop, the learning strategy corresponds to solve an optimization problem as presented in Part II of Algorithm 3.1, which is to minimize the difference (loss) of the two sides of the B-SPDE in (1.4) along each sample path in the squared error sense. In the learning iteration of (3.14), the learning rate α is designed to satisfy the condition $\alpha K_D^1 < 1$ for a constant K_D^1 (that is presented in (3.22) of this paper). It is worthwhile to point out that, even after we add this ML loop, we still can prove the convergence of this newly designed Algorithm 3.1, which is presented in the next subsubsection.

Finally, the associated simulation examples based on Algorithm 3.1 will be provided in Section 4. Instead, in the next subsection, we first conduct the convergence analysis with error bound estimation.

3.2. Convergence with error bound estimation

In this subsection, we focus on the discussion concerning the convergence and error bound estimation of Algorithm 3.1. More precisely, for each $t \in [0, T]$, $x \in \mathcal{X}$, and $j_0 \in \{n_0, n_0 - 1, ..., 1\}$, we define

$$\begin{aligned}
\Delta V_{\bar{K}}^{\epsilon,K}(t, x) &= V_{\bar{K}}^{\epsilon,K}(t, x) - V_\pi^\epsilon(t, x), && (3.17)\\
\Delta \bar{V}_{\bar{K}}^{\epsilon,K}(t, x) &= \bar{V}_{\bar{K}}^{\epsilon,K}(t, x) - \bar{V}_\pi^\epsilon(t, x), && (3.18)\\
V_\pi^\epsilon(t, x) &= V_\pi^\epsilon(t_{j_0-1}, x),\ t \in [t_{j_0-1}, t_{j_0}),\\
\bar{V}_\pi^\epsilon(t, x) &= \bar{V}_\pi^\epsilon(t_{j_0-1}, x),\ t \in [t_{j_0-1}, t_{j_0}).
\end{aligned}$$

Furthermore, let

$$\xi(k) = \sum_{i_0=1}^{k} 1 + \sum_{i_0=1}^{k}\sum_{i_1=1}^{i_0} 1 + \sum_{i_0=1}^{k}\sum_{i_1=1}^{i_0}\sum_{i_2=1}^{i_1} 1 + \cdots + \sum_{i_0=1}^{k}\sum_{i_1=1}^{i_0}\sum_{i_2=1}^{i_1}\cdots\sum_{i_{m-1}=1}^{2} 1. \tag{3.19}$$

In addition, it follows from the generalized local Lipschitz and linear growth conditions in (2.8) and (2.9) that

$$|\Delta(\nabla G)(s, x, u, v)| \leq K_D^1 \|u - v\|_{C^2(D,R)}, \tag{3.20}$$

$$|\nabla G(s,x,u))| \quad \leq \quad K_D^1 \|u\|_{C^2(D,R)} \tag{3.21}$$

for some positive constant K_D^1, where $(t,x) \in [0,T] \times D$ and $u, v \in C^2(D,R) \times C^2(D,R)$. Therefore, we can introduce the following assumption concerning the learning rate α,

$$\alpha K_D^1 < 1. \tag{3.22}$$

Finally, let $\|\cdot\|$ be the largest absolute value corresponding to each used function for all $x \in \mathcal{X}$. Then, we can present our algorithm convergence theorem with error bound estimation as follows.

Theorem 3.1. *For Algorithm 3.1 with the condition in (3.22) and a given iteration number k in (3.14), there is a nonnegative constant C depending only on the terminal time T, the Lipschitz constant K_D in (2.8) and (2.9), and the supremum (a constant) in (2.6) such that the following mean-square error estimation is true,*

$$\sup_{t\in[0,T]} \left(E\left[\left\| \Delta V_{\bar{K}}^{\epsilon,K}(t) \right\|^2 \right] + E\left[\left\| \Delta \bar{V}_{\bar{K}}^{\epsilon,K}(t) \right\|^2 \right] \right) \leq C|\pi| \tag{3.23}$$

for all sufficiently small $|\pi|$. In addition, consider a sequence of increasing sets $\mathcal{X}_{\bar{k}}$ with $\bar{k} \in \{1,2,...\}$ (i.e., $\mathcal{X}_1 \subset \mathcal{X}_2 \subset \cdots$). Suppose that the corresponding maximal mesh gauge $|\pi|_{\bar{k}}$ along $\bar{k} \in \{1,2,...\}$ satisfies

$$|\pi|_{\bar{k}} \to 0, \qquad \sum_{\bar{k}=1}^{\infty} \left(|\pi|_{\bar{k}} \right)^{\frac{1}{3}} < \infty.$$

Then, for a given $\mathcal{X} \in \{\mathcal{X}_1, \mathcal{X}_2, ...\}$, we have the a.s. convergence for Algorithm 3.1 as $\bar{k} \to \infty$,

$$\sup_{t\in[0,T]} \left(\left\| \Delta V_{\bar{K}}^{\epsilon,K}(t) \right\| + \left\| \Delta \bar{V}_{\bar{K}}^{\epsilon,K}(t) \right\| \right) \to 0 \quad \text{a.s.} \tag{3.24}$$

Proof. First, for the machine learning loop given by (3.14) in Part II of Algorithm 3.1, a given integer $k \in \{0,1,2,...\}$, and an index $j_0 \in \{n_0, n_0-1, ..., 1\}$, it follows from the facts in (3.20) and (3.21) that

$$\begin{aligned}
\left\| V_\pi^{\epsilon,k+1}(t_{j_0-1}) - V_\pi^{\epsilon,0}(t_{j_0-1}) \right\| \quad &\leq \quad \alpha \sum_{i_0=1}^{k} \left\| \nabla G(V_\pi^{\epsilon,i_0}(t_{j_0-1})) \right\| \qquad (3.25)\\
&\leq \quad \alpha \sum_{i_0=1}^{k} \left(\sum_{i_1=1}^{i_0} \left\| \nabla G(V_\pi^{\epsilon,i_1}(t_{j_0-1})) - \nabla G(V_\pi^{\epsilon,i_1-1}(t_{j_0-1})) \right\| + \left\| \nabla G(V_\pi^{\epsilon,0}(t_{j_0-1})) \right\| \right)\\
&\leq \quad \alpha|\pi| K_D^1 \sum_{i_0=1}^{k} \left(\sum_{i_1=1}^{i_0} \left\| V_\pi^{\epsilon,i_1}(t_{j_0-1}) - V_\pi^{\epsilon,i_1-1}(t_{j_0-1}) \right\|_{C^2(\mathcal{X},R)} + \left\| V^{\epsilon,0}(t_{j_0-1}) \right\|_{C^2(\mathcal{X},R)} \right)\\
&\leq \quad \alpha|\pi| K_D^1 \sum_{i_0=1}^{k} \left(\sum_{i_1=1}^{i_0} \alpha|\pi| K_D^1 \left(\sum_{i_2=1}^{i_1} \left\| V_\pi^{\epsilon,i_2}(t_{j_0-1}) - V_\pi^{\epsilon,i_2-1}(t_{j_0-1}) \right\|_{C^2(\mathcal{X},R)} \right.\right.\\
&\qquad \left.\left. + \left\| V_\pi^{\epsilon,0}(t_{j_0-1}) \right\|_{C^2(\mathcal{X},R)} \right) + \left\| V_\pi^{\epsilon,0}(t_{j_0-1}) \right\|_{C^2(\mathcal{X},R)} \right)\\
&\quad \cdots
\end{aligned}$$

$$\cdots$$
$$\cdots$$

$$\leq \alpha|\pi|K_D^1\xi(k)\left\|V_\pi^{\epsilon,0}(t_{j_0-1})\right\|_{C^2(\mathcal{X},R)},$$

where $\xi(k)$ is given in (3.19). Thus, for the given iteration number k and $t \in [t_{j_0-1}, t_{j_0})$, it follows from (3.25) and the discussion in Dai [9] that

$$\begin{aligned}
E\left[\left\|\Delta V_{\bar{K}}^{\epsilon,K}(t)\right\|^2\right] &= E\left[\left\|V_{\bar{K}}^{\epsilon,K}(t) - V_\pi^{\epsilon,k+1}(t)\right\|^2\right] \\
&\leq 2E\left[\left\|V_{\bar{K}}^{\epsilon,K}(t,x) - V_\pi^{\epsilon,0}(t)\right\|^2\right] + 2E\left[\left\|V_\pi^{\epsilon,0}(t) - V_\pi^{\epsilon,k+1}(t)\right\|^2\right] \\
&\leq C_1|\pi| + \left(\alpha|\pi|K_D^1\xi(k)\right)^2 E\left[\left\|V_\pi^{\epsilon,0}(t_{j_0-1})\right\|^2_{C^2(\mathcal{X},R)}\right] \\
&\leq C_1|\pi| + 2\left(\alpha|\pi|K_D^1\xi(k)\right)^2\left(E\left[\left\|V_\pi^{\epsilon,0}(t) - V_{\bar{K}}^{\epsilon,K}(t)\right\|^2_{C^2(\mathcal{X},R)}\right] + E\left[\left\|V_{\bar{K}}^{\epsilon,K}(t)\right\|^2_{C^2(\mathcal{X},R)}\right]\right) \\
&\leq C_1|\pi| + 2\left(\alpha|\pi|K_D^1\xi(k)\right)^2\left(C_1|\pi| + \sup_{t\in[0,T]} E\left[\left\|V_{\bar{K}}^{\epsilon,K}(t)\right\|^2_{C^2(D,R)}\right]\right),
\end{aligned} \tag{3.26}$$

where C_1 is some constant depending on T and K_D. Since $V_\pi^{\epsilon,0}(t_{j_0-1}, x)$ for each $j_0 \in \{n_0, n_0-1, ..., 1\}$ satisfies the equation in (3.12), it follows from Theorem 2.1 and (3.26) that the claim in (3.23) is true. Furthermore, the claim in (3.24) can be similarly proved by applying the discussion in Dai [9]. Hence, we finish the proof of Theorem 3.1. □

Now, if the B-SPDE in (1.1) has a $\{\mathcal{F}_t\}$-adapted solution pair $(V(t,x), \bar{V}(t,x))$, then $V_{xx}(t,x)$ can not be zero a.s. at all points $\{(t,x) \in [0,T]\times D\}$. Therefore, the numerical procedure in Algorithm 3.1 can also be applied to solve the equation in (1.1), e.g.,

$$V_\pi(t_{n_0}, x) = H_\pi(x), \tag{3.27}$$
$$\bar{V}_\pi(t_{n_0}, x) = 0, \tag{3.28}$$
$$V_\pi(t_{j_0-1}, x) = E\left[V_\pi(t_{j_0}, x) + \mathcal{L}_\pi(t_{j_0}, x, V_\pi(t_{j_0}, x))\Delta_{j_0}^{t,\pi}\middle|\mathcal{F}_{t_{j_0-1}}\right], \tag{3.29}$$
$$\begin{aligned}
\bar{V}_\pi(t_{j_0-1}, x) &= \frac{1}{\Delta_{j_0}^\pi}E\left[V_\pi(t_{j_0}, x)\Delta^\pi W_{j_0}\middle|\mathcal{F}_{t_{j_0-1}}\right] \\
&\quad + E\left[\mathcal{L}_\pi(t_{j_0}, x, V_\pi(t_{j_0}, x))\Delta^\pi W_{j_0}\middle|\mathcal{F}_{t_{j_0-1}}\right],
\end{aligned} \tag{3.30}$$

where $\mathcal{L}_\pi$ in (3.29) and (3.30) is the discrete version of the following partial differential operator,

$$\mathcal{L}(t,x,V,\bar{V}) = \frac{(V_x(t,x) + \bar{V}_x(t,x))^2}{V_{xx}(t,x)}. \tag{3.31}$$

Then, we have the following corollary.

Corollary 3.2. *Under the conditions in (3.22) with the Lipschitz constant K_D in (2.8)−(2.9), if there is a pair of $\{\mathcal{F}_t\}$-adapted solutions $(V(t,x), \bar{V}(t,x))$ in the space $\bar{\mathcal{Q}}^2_{\mathcal{F}}([0,T]\times D)$ to the B-SPDE in (1.1), then we have that*

$$\left(V_\pi^\epsilon(t_{j_0}, x), \bar{V}_\pi^\epsilon(t_{j_0}, x)\right) \to \left(V_\pi(t_{j_0}, x), \bar{V}_\pi(t_{j_0}, x)\right) \quad a.s. \tag{3.32}$$

$$E\left[\left|\left(V_\pi^\epsilon(t_{j_0},x),\bar{V}_\pi^\epsilon(t_{j_0},x)\right)-\left(V_\pi(t_{j_0},x),\bar{V}_\pi(t_{j_0},x)\right)\right|\right]\to 0 \tag{3.33}$$

as $\bar{K}\to\infty$ *first,* $K\to\infty$ *second,* $\epsilon\to 0$ *third for each* $x\in\mathcal{X}$ *and each* $j_0\in\{n_0,n_0-1,...,1\}$ *in a backward way, and* $|\pi|_{\bar{k}}\to 0$ *last along* $\bar{k}\in\{1,2,...\}$.

Proof. We prove the claim in (3.32) and (3.33) by induction in terms of $j_0\in\{n_0,n_0-1,...,2,1\}$. First, we consider the case that $j_0=n_0-1$, and it follows from (3.12) and (3.29) that

$$\begin{aligned}
&\left|V_\pi(t_{n_0-1},x)-V_\pi^\epsilon(t_{n_0-1},x)\right|\\
&\le E\left[\left(\frac{(V_x(t_{n_0},x))^2}{|V_{xx}(t_{n_0},x)|}+\frac{\bar{K}^2}{|V_{xx}(t_{n_0},x)|}\right)I_{\{\epsilon\le|V_{xx}(t_{n_0},x)|\le K\}}I_{\{|V_x(t_{n_0},x)|>\bar{K}\}}\Delta_{n_0}^{t,\pi}\middle|\mathcal{F}_{t_{n_0-1}}\right]\\
&+E\left[\left(\frac{(V_x(t_{n_0},x))^2}{|V_{xx}(t_{n_0},x)|}+\frac{\bar{K}^2}{\epsilon}\right)I_{\{|V_{xx}(t_{n_0},x)|<\epsilon\}}I_{\{|V_x(t_{n_0},x)|>\bar{K}\}}\Delta_{n_0}^{t,\pi}\middle|\mathcal{F}_{t_{n_0-1}}\right]\\
&\quad+E\left[\left(\frac{(V_x(t_{n_0},x))^2}{|V_{xx}(t_{n_0},x)|}+\frac{\bar{K}^2}{K}\right)I_{\{|V_{xx}(t_{n_0},x)|>K\}}I_{\{|V_x(t_{n_0},x)|>\bar{K}\}}\Delta_{n_0}^{t,\pi}\middle|\mathcal{F}_{t_{n_0-1}}\right]\\
&+E\left[\left(\frac{(V_x(t_{n_0},x))^2}{|V_{xx}(t_{n_0},x)|}+\frac{(V_x(t_{n_0},x))^2}{\epsilon}\right)I_{\{|V_{xx}(t_{n_0},x)|<\epsilon\}}I_{\{|V_x(t_{n_0},x)|\le\bar{K}\}}\Delta_{n_0}^{t,\pi}\middle|\mathcal{F}_{t_{n_0-1}}\right]\\
&+E\left[\left(\frac{(V_x(t_{n_0},x))^2}{|V_{xx}(t_{n_0},x)|}+\frac{(V_x(t_{n_0},x))^2}{K}\right)I_{\{|V_{xx}(t_{n_0},x)|>K\}}I_{\{|V_x(t_{n_0},x)|\le\bar{K}\}}\Delta_{n_0}^{t,\pi}\middle|\mathcal{F}_{t_{n_0-1}}\right]\\
&\le 2E\left[\frac{(V_x(t_{n_0},x))^2}{|V_{xx}(t_{n_0},x)|}I_{\{\epsilon\le|V_{xx}(t_{n_0},x)|\le K\}}I_{\{|V_x(t_{n_0},x)|>\bar{K}\}}\Delta_{n_0}^{t,\pi}\middle|\mathcal{F}_{t_{n_0-1}}\right]\\
&+2E\left[\frac{(V_x(t_{n_0},x))^2}{|V_{xx}(t_{n_0},x)|}I_{\{|V_{xx}(t_{n_0},x)|<\epsilon\}}I_{\{|V_x(t_{n_0},x)|>\bar{K}\}}\Delta_{n_0}^{t,\pi}\middle|\mathcal{F}_{t_{n_0-1}}\right]\\
&+\quad E\left[\left(\frac{(V_x(t_{n_0},x))^2}{|V_{xx}(t_{n_0},x)|}+\frac{(V_x(t_{n_0},x))^2}{K}\right)I_{\{|V_{xx}(t_{n_0},x)|>K\}}I_{\{|V_x(t_{n_0},x)|>\bar{K}\}}\Delta_{n_0}^{t,\pi}\middle|\mathcal{F}_{t_{n_0-1}}\right]\\
&+2E\left[\frac{(V_x(t_{n_0},x))^2}{|V_{xx}(t_{n_0},x)|}I_{\{|V_{xx}(t_{n_0},x)|<\epsilon\}}I_{\{|V_x(t_{n_0},x)|\le\bar{K}\}}\Delta_{n_0}^{t,\pi}\middle|\mathcal{F}_{t_{n_0-1}}\right]\\
&+\quad E\left[\left(\frac{(V_x(t_{n_0},x))^2}{|V_{xx}(t_{n_0},x)|}+\frac{(V_x(t_{n_0},x))^2}{K}\right)I_{\{|V_{xx}(t_{n_0},x)|>K\}}I_{\{|V_x(t_{n_0},x)|\le\bar{K}\}}\Delta_{n_0}^{t,\pi}\middle|\mathcal{F}_{t_{n_0-1}}\right]\\
&\le 2E\left[\frac{(V_x(t_{n_0},x))^2}{|V_{xx}(t_{n_0},x)|}I_{\{|V_x(t_{n_0},x)|>\bar{K}\}}\Delta_{n_0}^{t,\pi}\middle|\mathcal{F}_{t_{n_0-1}}\right]\\
&+2E\left[\frac{(V_x(t_{n_0},x))^2}{|V_{xx}(t_{n_0},x)|}I_{\{|V_x(t_{n_0},x)|>\bar{K}\}}\Delta_{n_0}^{t,\pi}\middle|\mathcal{F}_{t_{n_0-1}}\right]\\
&+\quad E\left[\left(\frac{(V_x(t_{n_0},x))^2}{|V_{xx}(t_{n_0},x)|}+\frac{(V_x(t_{n_0},x))^2}{K}\right)I_{\{|V_x(t_{n_0},x)|>\bar{K}\}}\Delta_{n_0}^{t,\pi}\middle|\mathcal{F}_{t_{n_0-1}}\right]\\
&+2E\left[\frac{(V_x(t_{n_0},x))^2}{|V_{xx}(t_{n_0},x)|}I_{\{|V_{xx}(t_{n_0},x)|<\epsilon\}}\Delta_{n_0}^{t,\pi}\middle|\mathcal{F}_{t_{n_0-1}}\right]\\
&+\quad E\left[\left(\frac{(V_x(t_{n_0},x))^2}{|V_{xx}(t_{n_0},x)|}+\frac{(V_x(t_{n_0},x))^2}{K}\right)I_{\{|V_{xx}(t_{n_0},x)|>K\}}\Delta_{n_0}^{t,\pi}\middle|\mathcal{F}_{t_{n_0-1}}\right].
\end{aligned}\tag{3.34}$$

Then, by the monotone (and conditional monotone) convergence theorem, we first let $\bar{K} \to \infty$, second let $K \to \infty$, and third let $\epsilon \to 0$, and we can conclude that

$$\left|V_\pi^\epsilon(t_{n_0-1}, x) - V_\pi(t_{n_0-1}, x)\right| \to 0 \quad \text{a.s.}, \tag{3.35}$$

$$E\left[\left|V_\pi^\epsilon(t_{n_0-1}, x) - V_\pi(t_{n_0-1}, x)\right|\right] \to 0. \tag{3.36}$$

Second, we consider the case that $j_0 = n_0 - 2$. Let $V_{\pi,\bar{K}}^{\epsilon,K}(t_{n_0-1}, x)$ be the corresponding value computed through (3.29). Then, it follows from (3.12) and (3.29) that

$$\begin{aligned}
&\left|V_\pi(t_{n_0-2}, x) - V_\pi^\epsilon(t_{n_0-2}, x)\right| \le E\left[\left|\Delta V_{\bar{K}}^{\epsilon,K}(t_{n_0-1}, x)\right| \middle| \mathcal{F}_{t_{n_0-2}}\right] + E\left[\left|V_\pi(t_{n_0-1}, x) - V_{\pi,\bar{K}}^{\epsilon,K}(t_{n_0-1}, x)\right| \middle| \mathcal{F}_{t_{n_0-2}}\right] \\
&+ E\left[\left(\frac{(V_x(t_{n_0-1}, x) + \bar{V}_x(t_{n_0-1}, x))^2}{|V_{xx}(t_{n_0-1}, x)|} + \frac{\bar{K}^2}{|V_{xx}(t_{n_0-1}, x)|}\right) I_{\{\epsilon \le |V_{xx}(t_{n_0-1}, x)| \le K\}} I_{\{|V_x(t_{n_0-1}, x) + \bar{V}_x(t_{n_0-1}, x)| > \bar{K}\}} \Delta_{n_0-1}^{t,\pi} \middle| \mathcal{F}_{t_{n_0-2}}\right] \\
&+ E\left[\left(\frac{(V_x(t_{n_0}, x) + \bar{V}_x(t_{n_0-1}, x))^2}{|V_{xx}(t_{n_0}, x)|} + \frac{\bar{K}^2}{\epsilon}\right) I_{\{|V_{xx}(t_{n_0}, x)| < \epsilon\}} I_{\{|V_x(t_{n_0-1}, x) + \bar{V}_x(t_{n_0-1}, x)| > \bar{K}\}} \Delta_{j_0}^{t,\pi} \middle| \mathcal{F}_{t_{n_0-2}}\right] \\
&\quad + E\left[\left(\frac{(V_x(t_{n_0-1}, x) + \bar{V}_x(t_{n_0-1}, x))^2}{|V_{xx}(t_{n_0-1}, x)|} + \frac{\bar{K}^2}{K}\right) I_{\{|V_{xx}(t_{n_0-1}, x)| > K\}} I_{\{|V_x(t_{n_0-1}, x) + \bar{V}_x(t_{n_0-1}, x)| > \bar{K}\}} \Delta_{n_0-1}^{t,\pi} \middle| \mathcal{F}_{t_{n_0-2}}\right] \\
&+ E\left[\left(\frac{(V_x(t_{n_0-1}, x) + \bar{V}_x(t_{n_0-1}, x))^2}{|V_{xx}(t_{n_0-1}, x)|} + \frac{(V_x(t_{n_0-1}, x) + \bar{V}_x(t_{n_0-1}, x))^2}{\epsilon}\right)\right. \\
&\quad \left. I_{\{|V_{xx}(t_{n_0-1}, x)| < \epsilon\}} I_{\{|V_x(t_{n_0-1}, x) + \bar{V}_x(t_{n_0-1}, x)| \le \bar{K}\}} \Delta_{n_0-1}^{t,\pi} \middle| \mathcal{F}_{t_{n_0-2}}\right] \\
&+ E\left[\left(\frac{(V_x(t_{n_0-1}, x) + \bar{V}_x(t_{n_0-1}, x))^2}{|V_{xx}(t_{n_0-1}, x)|} + \frac{(V_x(t_{n_0-1}, x) + \bar{V}_x(t_{n_0-1}, x))^2}{K}\right)\right. \\
&\quad \left. I_{\{|V_{xx}(t_{n_0-1}, x)| > K\}} I_{\{|V_x(t_{n_0-1}, x) + \bar{V}_x(t_{n_0-1}, x)| \le \bar{K}\}} \Delta_{n_0-1}^{t,\pi} \middle| \mathcal{F}_{t_{n_0-2}}\right] \\
&\le E\left[\left|\Delta V_{\bar{K}}^{\epsilon,K}(t_{n_0-1}, x)\right| \middle| \mathcal{F}_{t_{n_0-2}}\right] + E\left[\left|V_\pi(t_{n_0-1}, x) - V_{\pi,\bar{K}}^{\epsilon,K}(t_{n_0-1}, x)\right| \middle| \mathcal{F}_{t_{n_0-2}}\right] \\
&+ 4E\left[\frac{(V_x(t_{n_0-1}, x) + \bar{V}_x(t_{n_0-1}, x))^2}{|V_{xx}(t_{n_0-1}, x)|} I_{\{|V_x(t_{n_0-1}, x)| + \bar{V}_x(t_{n_0-1}, x) > \bar{K}\}} \Delta_{n_0-1}^{t,\pi} \middle| \mathcal{F}_{t_{n_0-2}}\right] \\
&+ 2E\left[\left(V_x(t_{n_0-1}, x) + \bar{V}_x(t_{n_0-1}, x)\right)^2 \left(\frac{1}{|V_{xx}(t_{n_0-1}, x)|} + \frac{1}{K}\right) I_{\{|V_{xx}(t_{n_0-1}, x)| > K\}} \Delta_{n_0-1}^{t,\pi} \middle| \mathcal{F}_{t_{n_0-2}}\right] \\
&+ 2E\left[\frac{(V_x(t_{n_0-1}, x) + \bar{V}_x(t_{n_0-1}, x))^2}{|V_{xx}(t_{n_0-1}, x)|} I_{\{|V_{xx}(t_{n_0-1}, x)| < \epsilon\}} \Delta_{n_0-1}^{t,\pi} \middle| \mathcal{F}_{t_{n_0-2}}\right].
\end{aligned} \tag{3.37}$$

Then, by the monotone convergence theorem, we first let $\bar{K} \to \infty$, second let $K \to \infty$, and third let $\epsilon \to 0$, and it follows from (3.35) and (3.36) that

$$\begin{aligned}
&\left|V_\pi^\epsilon(t_{n_0-2}, x) - V_\pi(t_{n_0-2}, x)\right| \\
\to\ & E\left[\left|\Delta V_{\bar{K}}^{\epsilon,K}(t_{n_0-1}, x)\right| \middle| \mathcal{F}_{t_{n_0-2}}\right] + E\left[\left|V_\pi(t_{n_0-1}, x) - V_{\pi,\bar{K}}^{\epsilon,K}(t_{n_0-1}, x)\right| \middle| \mathcal{F}_{t_{n_0-2}}\right] \quad \text{a.s.},
\end{aligned} \tag{3.38}$$

$$E\left[\left|V_\pi^\epsilon(t_{n_0-2},x)-V_\pi(t_{n_0-2},x)\right|\right] \tag{3.39}$$
$$\to \quad E\left[\left|\Delta V_{\bar{K}}^{\epsilon,K}(t_{n_0-1},x)\right|\right]+E\left[\left|V_\pi(t_{n_0-1},x)-V_{\pi,\bar{K}}^{\epsilon,K}(t_{n_0-1},x)\right|\right].$$

Finally, a similar argument can be applied to the analysis for $(\bar{V}_\pi(t_{n_0-1},x)-\bar{V}_\pi^\epsilon(t_{n_0-1},x))$. Then, it follows from the induction, dominated convergence theorem, and Theorem 3.1 that the claims in (3.32) and (3.33). Hence, we have finished the proof of Corollary 3.2. □

4. Simulation examples

In this section, we present simulation examples to show the effectiveness of Algorithm 3.1. More precisely, we first simulate the paired solution to the approximated B-SPDE in (1.4). Since there is no analytic solution available to the B-SPDE, we conduct numerical comparison concerning the difference between two sides of our simulated B-SPDE to show the correctness of our computed solution. All of the results concerning the simulated solution are presented in Figures 2, 6, and 8. Second, based on the simulated solution of the B-SPDE, we also conduct further simulations concerning the financial investment strategies derived in (2.16). All of the simulated results are displayed in Figures 3–5. Note that some predicted properties in theory concerning the financial investment strategies are found in the simulation results, which further justifies the correctness of our simulated solution to the B-SPDE in (1.4).

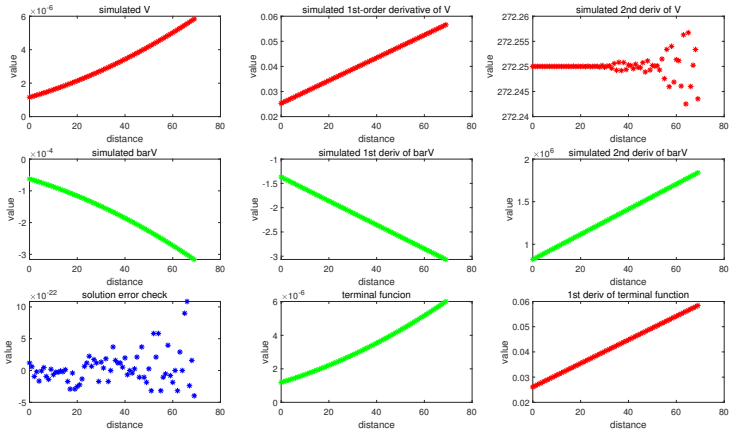

Figure 6. Simulated solution pair and pathwise error comparison to the B-SPDE in (1.4) with $\epsilon = 1/100000$, $K = upperbound = 10^9$, $\bar{K} = 2^{64}$, and $learningrate = 1/2$. Furthermore, $T = 0.1$, $n = 60000$, $hhh = 2$, $terminalcoefficient = 2 * 100$, $Q = 2352$, $b = 1/6000$, $d = 100$, $dropnum = 30$, $BMD = 2$, $bmdp = 3000$, and $k = 1$. In this figure, we display the evolution results of $(d - dropnum)$ points with respect to the position parameter $x \in \{b/d, 2b/d, ..., (d - dropnum)b/d\}$ over horizon-axes at a particular time point $T(n-8)/n$.

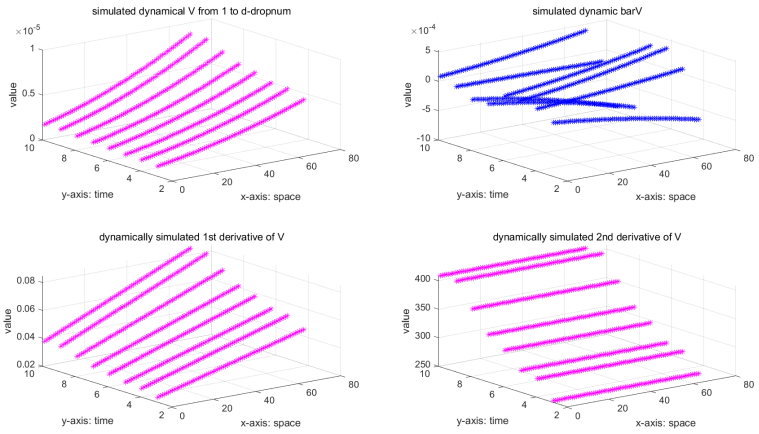

Figure 7. First case of the simulated dynamic evolving solution pair to the B-SPDE in (1.4) with $\epsilon = 1/100000$, $K = upperbound = 10^9$, $\bar{K} = 2^{64}$, and $learningrate = 1/2$. Furthermore, $T = 0.1$, $n = 60000$, $hhh = 2$, $terminalcoefficient = 2 * 100$, $Q = 2352$, $b = 1/6000$, $d = 100$, $dropnum = 30$, $BMD = 2$, $bmdp = 3000$, and $k = 1$. In this figure, we display the evolution results of $(d - dropnum)$ points with respect to the position parameter $x \in \{b/d, 2b/d, ..., (d - dropnum)b/d\}$ over horizon-axes and time parameter $t \in \{T, T(n-1)/n, ..., T(n-8)/n\}$.

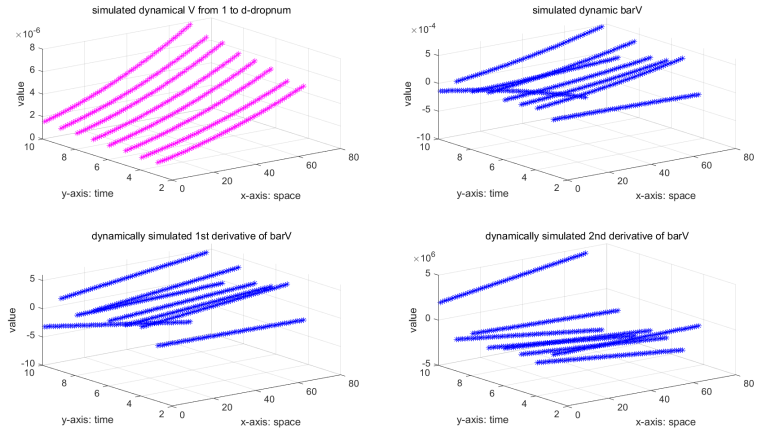

Figure 8. Second case of the simulated dynamic evolving solution pair to the B-SPDE in (1.4) with $\epsilon = 1/100000$, $K = upperbound = 10^9$, $\bar{K} = 2^{64}$, and $learningrate = 1/2$. Furthermore, $T = 0.1$, $n = 60000$, $hhh = 2$, $terminalcoefficient = 2 * 100$, $Q = 2352$, $b = 1/6000$, $d = 100$, $dropnum = 30$, $BMD = 2$, $bmdp = 3000$, and $k = 1$. In this figure, we display the evolution results of $(d - dropnum)$ points with respect to the position parameter $x \in \{b/d, 2b/d, ..., (d - dropnum)b/d\}$ over horizon-axes and time parameter $t \in \{T, T(n-1)/n, ..., T(n-8)/n\}$.

In all of the simulations, we use the notation T to denote the terminal time, a positive integer n to

denote the number of equally divided subintervals over $[0, T]$, a positive integer *hhh* to represent the highest order of partial derivatives corresponding to the implemented equation, a positive integer Q to be the total number of normally distributed random numbers, a positive number b to be the size for the space parameter, and a positive integer d to be the number of equally divided subintervals over $[0, b]$. Furthermore, we use a positive number BMD to denote the upper bound of an interval such that our driving Brownian motion $W \in [-BMD, BMD]$ and use a positive integer $bmdp$ to denote the number of equally divided subintervals over $[0, BMD]$. In addition, we use an integer $(n_0 + 1)$ to denote the number of layers of backward network as in the left part of Figure 1 and use an integer $(k+1)$ to denote the number of layers of reinforcement iterations as in the right part of Figure 1.

Note that the main purpose to impose the upper bounds K and $\bar{K}$ in (1.4) is for the equation in (1.4) to satisfy the general local Lipschitz and linear growth conditions. In our simulations, we take them to be sufficiently large in order that our simulation results are stable and acceptable. Actually, we take $\bar{K}$ to be the largest number allowed by our used computer, i.e., $\bar{K} = 2^{64}$. The computed first-order derivatives of $(V^{\epsilon,K}_{\bar{K}}, \bar{V}^{\epsilon,K}_{\bar{K}})$ are much less than this upper bound $\bar{K}$. In this sense, our simulations are stable and acceptable. Concerning K, we have used different numbers for tests. Our simulation results presented in this paper correspond to $K = 10^9$. If we used $K = 10^7$, the simulation results were not affected by this change. For safety, we used the simulation results corresponding to $K = 10^9$ in this paper.

However, since we are handling a B-SPDE in (1.1) with singularity, the choice of ϵ indeed has a significant impact on our simulation results. After careful tests, we chose to present the simulation results corresponding to $\epsilon = 1/100000$ in this paper. If we chose $\epsilon > 1/100000$ up to $\epsilon = 1/100$, the corresponding simulation results were still reasonable and acceptable. Nevertheless, if we chose $\epsilon < 1/100000$ significantly, our simulation results were not acceptable. Concerning the choice of the learning rate α, we chose $\alpha = 1/2$ for good simulation results. The simulation results corresponding to α around 1/2 were also acceptable. Nevertheless, if we chose $\alpha = 1$, the simulation results are not acceptable.

In Figure 2, we present the simulation results of the paired solution $(V^{\epsilon,K}_{\bar{K}}, \bar{V}^{\epsilon,K}_{\bar{K}})$ to the approximated B-SPDE in (1.4) with $\epsilon = 1/100000$, $K = 10^9$, and $\bar{K} = 2^{64}$. The computed values at time point $t_{j_0} = t_{n_0}$ with $n_0 = 10$ correspond to the input terminal values at $t_{n_0+1} = T$. The terminal random field $H(x)$ in (1.4) is taken to be the form

$$H(x) = C(1 + W^2(T))\left(x + \frac{5.5}{60000}\right)^2, \tag{4.1}$$

where C is a positive constant and is taken to be $terminalcoefficient = 2 * 100$ as in the explanation for Figure 2. Note that the "solution error check" titled in the third plot of the first column in Figure 2 is in terms of the difference (denoted by "Err") between the two sides of the B-SPDE in (1.4) and is with respect to a particular sample path. In applying Algorithm 3.1 with the terminal time at $t_{n_0+1} = T$ to solve the equation in (1.4), the corresponding Err at time t_{n_0} has the following expression,

$$\begin{aligned} \mathrm{Err}(t_{n_0}, x) &= V(t_{n_0}, x) - H(x) + \frac{1}{2}\Big(T - t_{n_0}\Big)\frac{\Phi_{\bar{K}}(V_x(t_{n_0}, x) + \bar{V}_x(t_{n_0}, x))}{\Psi^{\epsilon,K}(V_{xx}(t_{n_0}, x)} \\ &\quad + \Big(W(T) - W(t_{n_0})\Big)\bar{V}(t_{n_0}, x). \end{aligned}$$

From the simulation results displayed in the third plot of the first column, we can see that our algorithm is quite accurate. Furthermore, the three plots in the first row of Figure 2 display the

simulated $V_{\bar{K}}^{\epsilon,K}(t_{n_0}, x)$ together with its simulated first-order and second-order derivatives in terms of position parameter $x \in \{b/d, 2b/d..., (d - dropnum)b/d\}$ with $d = 100$ and $dropnum = 30$. Although the graph in the third plot is non-smooth, it is close to a smooth line. In addition, the three plots in the second row of Figure 2 display the simulated $\bar{V}_{\bar{K}}^{\epsilon,K}(t_{n_0}, x)$ together with its simulated first-order and second-order derivatives. Finally, the second and third plots in the third row of Figure 2 display the simulated terminal value $V_{\bar{K}}^{\epsilon,K}(t_{n_0+1}, x)$ together with its simulated first-order derivative. In Figure 6, we display the similar plots as in Figure 2 but with the simulated results at time point $t_{n_0-8} = t_3$.

In Figure 7, we display the first case of the solution evolving as the time index j_0 decreases from $n_0 + 1$ to 3 with $n_0 = 10$. More precisely, the three graphs in the magenta color display the dynamic evolving of $V_{\bar{K}}^{\epsilon,K}(t_{j_0}, x)$ together with their first-order and second-order derivatives. The graph in the blue color displays the solution evolving of $\bar{V}_{\bar{K}}^{\epsilon,K}(t_{j_0}, x)$. Similarly, in Figure 8, we display the second case of the solution evolving as the time index j_0 decreases from $n_0 + 1$ to 3 with $n_0 = 10$. More precisely, the three graphs in the blue color display the dynamic evolving of $\bar{V}_{\bar{K}}^{\epsilon,K}(t_{j_0}, x)$ together with their first-order and second-order derivatives. The graph in the magenta color displays the solution evolving of $V_{\bar{K}}^{\epsilon,K}(t_{j_0}, x)$.

In Figure 3, we display the simulated investment policy, myopic policy, and excess hedging demand at time point t_{j_0} with $j_0 = n_0$ and $n_0 = 10$. These policies correspond to the formula in (2.16) and its related explanations. The three graphs in the left column correspond to the simulated pathwise results. The three graphs in the right column correspond to the simulated results in the mean average sense with respect to the simulation iteration number Q. Theoretically, the myopic policy should continue to be constant. Our simulated results support this theoretical result. However, this theoretic result also further justifies the correctness of our algorithm and simulations. Similarly, in Figure 4, we display the simulated investment policy, myopic policy, and excess hedging demand at time point $t_{n_0-8} = t_3$ with $n_0 = 10$. Finally, in Figure 5, we display the simulated dynamical evolutions of investment policy, myopic policy, and excess hedging demand with respect to time parameter $t \in \{T, T(n-1)/n, ..., T(n-8)/n\}$.

5. Conclusions

In this paper, we studied a strongly nonlinear B-SPDE through an approximation method. This equation is well-known and was previously derived from studies in finance. However, how to analyze and solve this equation has continued to an open problem for quite a long time. Therefore, by applying our previously established theory and numerical scheme together with CNN and ML, we have developed an effective approximation method with a Monte Carlo simulation algorithm to tackle the well-known open problem. In doing so, the existence and uniqueness of the 2-tuple adapted strong solution to an approximation B-SPDE were proved. Meanwhile, the convergence of a newly designed simulation algorithm was established. Simulation examples and applications in finance were also provided.

Use of AI tools declaration

The author declares he has not used Artificial Intelligence (AI) tools in the creation of this article.

Acknowledgments

The author acknowledges that the project was funded by the National Natural Science Foundation of China with Grant No. 11771006. The author would like to thank the editors and the reviewers for their helpful comments and suggestions to revise this paper.

Conflict of interest

The author declares that he has no competing interests.

References

1. J. Braun, M. Griebel, On a constructive proof of Kolmogorov's superposition thoerem, *Constr. Approx.*, **35** (2009), 653–675. https://doi.org/10.1007/s00365-009-9054-2
2. A. Cěrný, J. Kallsen. On the structure of general mean-variance hedging strategies, *Ann. Appl. Probab.*, **35** (2007), 1479–1531. https://doi.org/10.1214/009117906000000872
3. G. Cybenko, Approximation by superpositions of a sigmoidal function, *Math. Control Signal System*, **1** (1989), 303–314. https://doi.org/10.1007/BF02551274
4. W. Dai, Brownian approximations for queueing networks with finite buffers: modeling, heavy traffic analysis and numerical implementations, Ph.D thesis, *Georgia Institute of Technology*, 1996.
5. J. G. Dai, W. Dai, A heavy traffic limit theorem for a class of open queueing networks with finite buffers, *Queueing Syst.*, **32** (1999), 5–40. https://doi.org/10.1023/A:1019178802391
6. W. Dai, Mean-variance portfolio selection based on a generalized BNS stochastic volatility model, *Int. J. Comput. Math.*, **88** (2011), 3521–3534. https://doi.org/10.1080/00207160.2011.606904
7. W. Dai, Optimal rate scheduling via utility-maximization for J-user MIMO Markov fading wireless channels with cooperation, *Oper. Res.*, **61** (2013), 1450–1462. https://doi.org/10.1287/opre.2013.1224
8. W. Dai, Mean-variance hedging based on an incomplete market with external risk factors of non-Gaussian OU processes, *Math. Probl. Eng.*, **2015** (2015), 625289. https://doi.org/10.1155/2015/625289
9. W. Dai, Convolutional neural network based simulation and analysis for backward stochastic partial differential equations, *Comput. Math. Appl.*, **119** (2022), 21–58. https://doi.org/10.1016/j.camwa.2022.05.019
10. W. Dai, Optimal policy computing for blockchain based smart contracts via federated learning, *Oper. Res. Int. J.*, **22** (2022), 5817–5844. https://doi.org/10.1007/s12351-022-00723-z
11. L. Gonon, L. Grigoryeva, J. P. Ortega, Approximation bounds for random neural networks and reservoir systems, *Ann. Appl. Probab.*, **33** (2023), 28–69. https://doi.org/10.1214/22-AAP1806
12. R. Gozalo-Brizuela, E. C. Garrido-Merchan, ChatGPT is not all you need. A state of the art review of large generative AI models, preprint paper, 2023. https://doi.org/10.48550/arXiv.2301.04655
13. S. Haykin, *Neural networks: A Comprehensive Foundation*, New Jersey: Prentice Hall PTR, 1994.
14. K. Hornik, M. Stinchcombe, H. White, Multilayer feedforward networks are universal approximators, *Neur. Networks*, **2** (1989), 359–366. https://doi.org/10.1016/0893-6080(89)90020-8

15. N. Ikeda, S. Watanabe, *Stochastic Differential Equations and Diffusion Processes*, 2 Eds., Kodansha: North-Holland, 1989.
16. O. Kallenberg, *Foundation of Modern Probability*, Berlin: Springer, 1997.
17. A. N. Kolmogorov, On the representation of continuous functions of several variables as superpositions of continuous functions of a smaller number of variables, *Dokl. Akad. Nauk*, **108** (1956).
18. D. Kramkov, M. Sirbu, On the two times differentiability of the value function in the problem of optimal investment in incomplete markets, *Ann. Appl. Probab.*, **16** (2006), 1352–1384. https://doi.org/10.1214/105051606000000259
19. A. Kratsios, V. Debarnot, I. Dokmannić, Small transformers compute universal metric embeddings, *J. Mach. Learning Res.*, **24** (2023), 1–48.
20. Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, et al., Backpropagation applied to handwritten zip code recognition, *Neur. Comput.*, **1** (1989), 541–551. https://doi.org/10.1162/neco.1989.1.4.541
21. Z. Liu, Y. Wang, S. Vaidya, F. Ruehle, J. Halverson, M. Soljačić, et al., KAN: Kolmogorov-Arnold networks, preprint paper, 2024. https://arxiv.org/pdf/2404.19756
22. M. Musiela, T. Zariphopoulou. Stochastic partial differential equations and portfolio choice, In: *Contemporary Quantitative Finance*, Berlin: Springer, 2009. https://doi.org/10.1007/978-3-642-03479-4_11
23. B. Øksendal, *Stochastic Differential Equations*, 6 Eds, New York: Springer, 2005.
24. B. Øksendal, A. Sulem, T. Zhang, A stochastic HJB equation for optimal control of forward-backward SDEs, In: *The Fascination of Probability, Statistics and their Applications*, Berlin: Springer, 2016.
25. S. Peluchetti, Diffusion bridge mixture transports, Schrödinger bridge problems and generative modeling, *J. Mach. Learning Res.*, **2**4 (2023), 1–51.
26. J. Sirignano, K. Spiliopoulos, Dgm: a deep learning algorithm for solving partial differential equations, *J. Comput. Phys.*, **375** (2018), 1339–1364. https://doi.org/10.1016/j.jcp.2018.08.029
27. A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, et al., Attention is all you need, *Adv. Neur. Informa. Proc. Syst.*, **30** (2017), 5998–6008.
28. R. Yamashitza, M. Nishio, R. K. G. Do, Togashi, Convolutional neural networks: an overview and application in radiology, *Insights into Imaging*, **9** (2018), 611–629. https://doi.org/10.1007/s13244-018-0639-9

AIMS Press