



---

*Research article*

## CATL's stock price forecasting and its derived option pricing: a novel extended fNSDE-net method

Xiao Qi<sup>1</sup>, Tianyao Duan<sup>1</sup>, Lihua Wang<sup>2</sup> and Huan Guo<sup>1,\*</sup>

<sup>1</sup> School of Artificial Intelligence, Jiangnan University, Wuhan 430056, China

<sup>2</sup> School of Medicine, Jiangnan University, Wuhan 430056, China

\* **Correspondence:** Email: [guohuan@jhun.edu.cn](mailto:guohuan@jhun.edu.cn).

**Abstract:** This paper presents a novel numerical method, named extended fractional neural stochastic differential equation (fNSDE)-Net, which combines the generative adversarial network (GAN) and fNSDE with a self-attention module. The method is designed to generate and forecast the stock price of Contemporary Amperex Technology Co., Ltd. (CATL) in China. The primary challenge of this study lies in the fact that the input consists of a single, irregular time-series dataset with long-range dependencies (i.e., Hurst index  $H > \frac{1}{2}$ ), and its inherent noise cannot be directly modeled using pure Brownian motion. The proposed method not only generates multiple sample paths based on the initial data in a probabilistic sense but also preserves the long-term memory characteristics of the generated samples. Moreover, the pricing of a Bermuda call option, induced by stock prices, is explored. Through a series of numerical error comparisons and estimator reliability tests, the proposed method outperforms both the pure fNSDE-GAN method and the NSDE-GAN method in terms of fitting and generalization performance, thereby demonstrating its effectiveness.

**Keywords:** extended fNSDE-Net; stock price generation/prediction; Bermuda option pricing; long-range dependency; CATL; Hurst index

**Mathematics Subject Classification:** 62M45, 62P05, 91G80

---

### 1. Introduction

Contemporary Amperex Technology Co., Ltd. (CATL), as a leading new energy innovation and technology company in China and even the world, has attracted widespread attention for its stock price [1–3]. Predicting the trend of stock price is an important basis for many investors to make investments. However, influenced by factors such as new energy costs, usability, economic conditions, and policy environments, CATL's stock price exhibits prolonged volatility and long-memory characteristics, which brings inherent difficulties to the prediction of its price trend. The

objective of this study is to design an efficient numerical method, named extended fractional neural stochastic differential equation (fNSDE)-Net, for simulating and forecasting stock prices and their corresponding Bermuda options, which exhibit long-term dependence characteristics. The core idea behind this method is to integrate the generative adversarial network (GAN) with the fNSDE and self-attention modules to generate and predict the prices of underlying assets. The main contributions of this paper are summarized as follows: The proposed method, extended fNSDE-Net, combines the strengths of fNSDE-Net, which excels at simulating long-range correlated time series, and the generative capabilities of GAN, enabling the generation of multiple simulation paths for a single stock price time series in a probabilistic sense. In particular, the self-attention mechanism is incorporated into the generator component of our method to better capture long-range dependencies. The results demonstrate that, compared to the pure fNSDE-GAN method [4] and the NSDE-GAN method [5], our approach outperforms them in terms of both fitting and generalization performance for stock prices exhibiting temporal memory effects. In addition to fitting/forecasting the stock price of CATL, we also address the pricing of a vanilla Bermuda call option derived from the generated stock price paths. Motivated by the least squares Monte (LSM) Carlo method originally proposed by Longstaff et al. [6], we apply dynamic programming principles to price the corresponding option. To the best of our knowledge, no existing work has combined fNSDE-Net with LSM methods for option pricing. This paper aims to make a modest contribution in this regard.

In fact, with the advancement of artificial intelligence (AI), the use of AI to simulate, calculate, and predict changes in new energy stock prices has become increasingly popular and has become a major subject of exploration for financial analysts and academic researchers. In recent years, some scholars have made significant contributions in this field, for example, Sadorsky [7] made use of the machine learning method of random forests to predict the stock price direction of clean energy exchange traded funds, and showed that random forests are a useful method for forecasting the clean energy stock price trends. In addition, Sadorsky [8] also used tree-based machine learning methods to forecast the direction of solar stock prices. Wang et al. [9] developed a novel prediction method by the integration of time series and cloud models to manage the stock price prediction of new energy vehicles enterprises. Gu et al. [10] proposed a neural network prediction method via Markov chain and long short-term memory (LSTM) neural network to predict new energy stock price, and proved that the new method was better than LSTM neural network in the evaluation indexes of average relative error, a posteriori error, mean square error and small probability error of stock price prediction. Meng et al. [11] employed a novel attention mechanism to forecast new energy electricity price, and demonstrated that the proposed model was superior to some hybrid models. Shen et al. [12] utilized autoregressive univariate time series models to predict the trends of new energy stock closing prices, and showed by comparison that the temporal convolutional attention neural network has a great effect on stock price prediction. Zhu et al. [13] introduced the new model called “VMD-BIGRU” to predict the trend of the portfolio and quantify the trades of the portfolios, and proved that this model greatly improved forecasting performance compared to single gated recurrent unit or LSTM models. Fan et al. [14] used a new AI model to predict the stock price of new energy industry. This method involved support vector machines and Ant Lion Optimization to accurately predict stock prices. Alshater et al. [15] deployed Machine Learning models to forecast energy equity prices by employing uncertainty indices as a proxy for predicting energy market volatility. Gao et al. [16] introduced the most commonly used machine learning techniques and explored their diverse applications in

marketing, stock analysis, demand forecasting, and energy marketing. Ghosh et al. [17] investigated the predictability of clean energy investment in the US market by using NeuralProphet and selecting eight sectoral stock indices. Jiang et al. [18] compared the application of various artificial intelligence methods in electricity price forecasting. Lin et al. [19] proposed a novel decomposition-ensemble model to predict the energy prices which fluctuate wildly. Mostafa et al. [20] used evolutionary techniques such as gene expression programming and artificial neural network models to predict oil prices over the period from January 2, 1986 to June 12, 2012. Pino et al. [21] predicted the hourly energy prices for the next-day in the Spanish electricity production market based on artificial neural networks and compared the calculation results with the ARIMA model. Although great progress has been made in numerical simulation and prediction of new energy stock prices using neural networks or machine learning methods, the problem of predicting the price of a single new energy stock with long-term memory effects and non-Brownian motion characteristics has not yet been addressed to the best of our knowledge. This paper attempts to fill this gap from the perspective of numerical approximation.

The rest of the paper is organized as follows: Section 2 theoretically outlines the details of the extended fNSDE-Net approach. Section 3 presents a series of numerical experiments to illustrate the effectiveness of the new approach. Section 4 introduces the pricing problem of a vanilla Bermuda call option induced by the generated stock price path and verifies the reliability of the pricing method through comparative experiments. Finally, Section 5 concludes.

## 2. Stock price and new network model

In this section, we first present the closing prices of CATL stock over the past year, revealing its long-range dependency characteristic. Then we introduce the proposed new method in detail. It is started by stating some basic mathematical concepts and definitions.

**Definition 2.1.** Let  $H \in (0, 1)$  be a fixed number. A real-valued mean-zero Gaussian process  $B^H(t), t \geq 0$ , is called fractional Brownian motion (fBm) with Hurst index  $H$  if it fulfills

$$B^H(0) = 0$$

almost surely and its covariance function satisfies:

$$\text{Cov}(B^H(s), B^H(t)) = \frac{1}{2}(|t|^{2H} + |s|^{2H} - |t - s|^{2H}), \quad s, t \geq 0. \quad (2.1)$$

Obviously, fBm becomes the pure Brownian motion when

$$H = \frac{1}{2}.$$

Moreover, fBm has stationary increments since

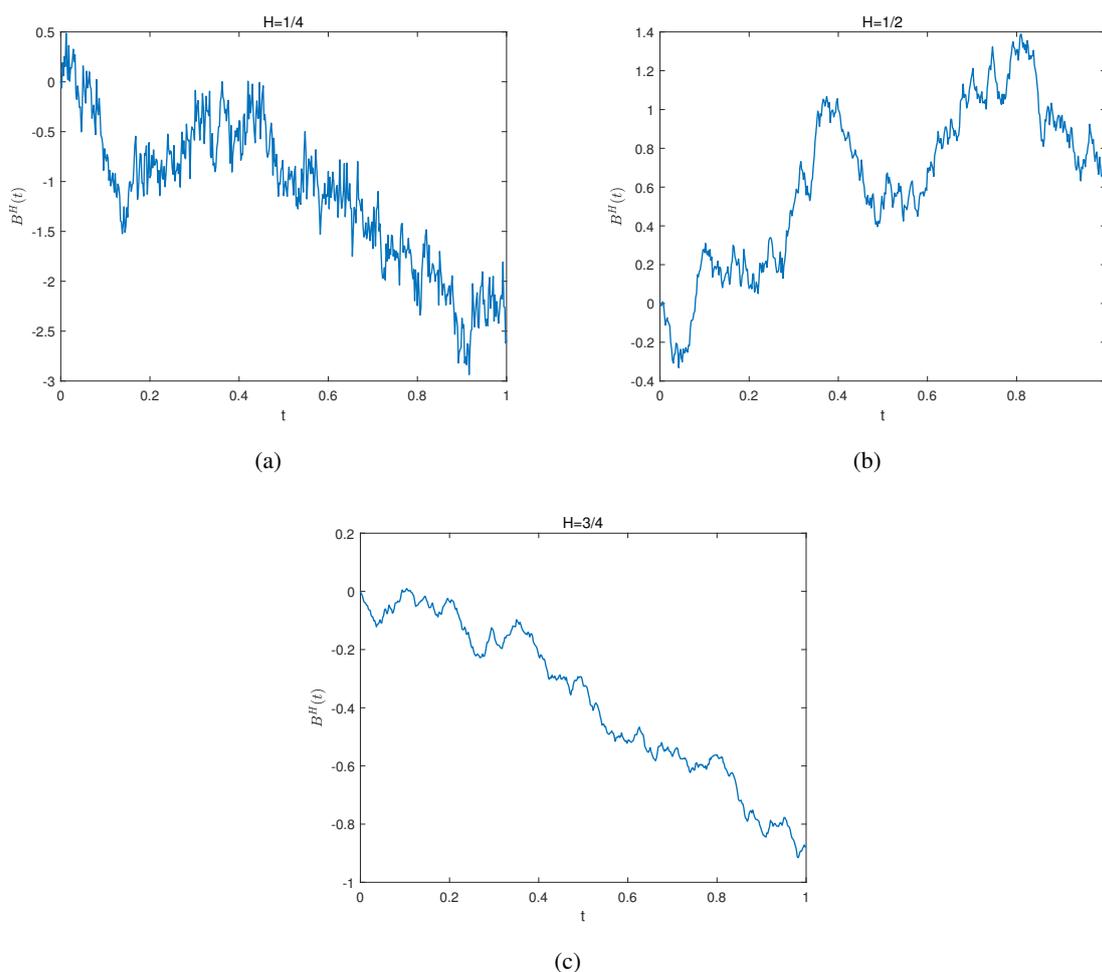
$$B^H(t) - B^H(s) \sim N(0, |t - s|^{2H}),$$

see, e.g., [22, Section 5.3] for detail. Especially, almost all paths of fBm with Hurst index  $H$  satisfy  $(H - \varepsilon)$ -Hölder regularity [23] for any  $\varepsilon > 0$ , i.e.,

$$|B^H(t) - B^H(s)| \leq c|t - s|^{H-\varepsilon}$$

for a constant  $c > 0$ , and  $s, t \geq 0$ . This means: sample path of fBm shows better regularity as the Hurst index  $H$  becomes large, as shown in Figure 1. It is worthwhile to point that according to the definition of long-range dependency [4, Definition III.2], one verifies readily that the increments of fBm exhibit long-range dependency if and only if

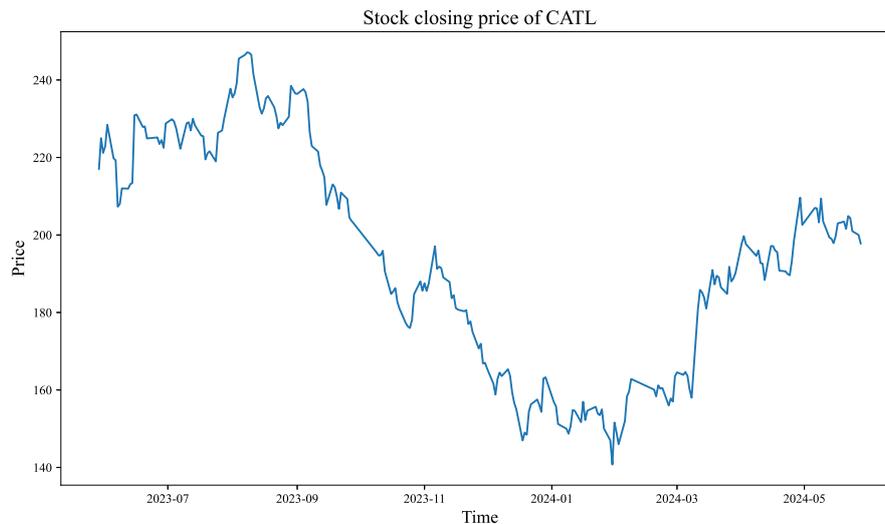
$$H > \frac{1}{2}.$$



**Figure 1.** Sample paths of fBm with different  $H$ ; (a):  $H = 1/4$ , (b):  $H = 1/2$ , (c):  $H = 3/4$ .

Next, we investigate the mathematical characteristic of stock price of CATL. Through AKShare database (see the link <https://github.com/akfamily/akshare>), we take the closing prices of CATL stock for all workdays from May 2023 to May 2024, which is given in Figure 2.

Indeed, AKShare is a Python library that provides financial data extraction and analysis tools for Chinese financial markets. Many researchers have made use of the data from AKShare to analyze stock price trends, see, e.g., [24, 25] and references therein.



**Figure 2.** CATL's closing price path for all workdays from May 2023 to May 2024.

Now we will show that the path of CATL's stock closing price is characterized by long-range dependency. To this end, we need to introduce the notion of R/S statistic.

Let

$$\{V(t), t = 0, 1, \dots, T\}$$

be a discrete random time series. Define the range of  $V(t)$  at time  $T$  by

$$R_T := \max_{0 \leq i \leq T} \sum_{t=0}^i (V(t) - \bar{V}(T)) - \min_{0 \leq i \leq T} \sum_{t=0}^i (V(t) - \bar{V}(T))$$

with

$$\bar{V}(T) := \frac{1}{T} \sum_{0 \leq t \leq T} V(t)$$

representing the sample mean. Furthermore, let  $S_T$  be the sample standard deviation defined by

$$S_T := \left( \frac{1}{T} \sum_{t=0}^T (V(t) - \bar{V}(T))^2 \right)^{\frac{1}{2}}.$$

The quantity  $R_T/S_T$  is called R/S statistics. An important application of R/S statistics is to determine whether a random time series  $V(t)$  can be simulated using fBm. To be specific, Mandelbrot [26] has proven that R/S statistics asymptotically satisfy the relation

$$\log(R_T/S_T) = H \log T + c + \mathcal{O}_T(1), \quad (2.2)$$

where  $H$  is the Hurst index,  $c$  is a constant, and  $\mathcal{O}_T(1)$  is a quantity that vanishes in probability as  $T$  tends to infinity. In fact, Eq (2.2) provides a natural way to estimate  $H$ , such as using a linear regression approach.

By simple calculation, we obtain the Hurst index that matches the closing price path of CATL stock is 0.721, which is bigger than 0.5, indicating that the increments of CATL stock price have long-range dependency. This implies that employing a dynamic process driven by fBm with

$$H = 0.721$$

to simulate the CATL stock price will be reasonable. In addition to the Hurst index of CATL's stock price, we also calculated the Hurst index for the stock prices of several other well-known companies (see Table 1 below for details) and found that, with the exception of CATL, the Hurst index for the stock prices of the others is less than  $1/2$ . In the current work, we focus on the stock price valuation problem with long-term dependencies (i.e.,  $H > 1/2$ ), and thus, we selected CATL's stock price as the underlying asset.

**Table 1.** Hurst index of stock prices of different companies.

Stock name	Hurst index
Trina Solar	0.374
Longi Green Energy	0.439
Hikvision	0.450
Industrial and Commercial Bank of China	0.471
Kweichow Moutai	0.479
Tongwei Co., Ltd.	0.486
Mindray Medical	0.493
<b>CATL</b>	<b>0.721</b>

One more thing to note is that CATL's stock price is just one path, and thus, there is a need to find a powerful generative model capable of producing more possible realizations in a probabilistic sense, thereby providing more valuable information for predicting stock prices. For this purpose, to address this, we propose a novel method, hereafter referred to as extended fNSDE-Net.

### 2.1. Extended fNSDE-Net

The extended fNSDE-Net consists of two main components called the generator and the discriminator. The generator is formulated by an NSDE driven by a fBm with a Hurst index of

$$H = 0.721,$$

which is used to generate multiple latent paths with long-range dependency. The discriminator is represented as a neural controlled differential equation (NCDE) by using the Riemann-Stieltjes measure of the generator output, aiming to provide a score of how close the generated path is to the true stock price path. Particularly, the self-attention module is embedded in the generator to efficiently model relationships between widely separated data. It is followed by a description of the generator.

**Generator.** Given a time horizon  $T > 0$ , let  $Y_{\text{true}}(t)$  denote the closing price of CATL stock,  $t \in [0, T]$ .  $Y_{\text{true}}(t)$  is what we wish to approximate, and it is a continuous stochastic process. Let

$$B^*(t) : [0, T] \rightarrow \mathbb{R}^w$$

be a  $w$ -dimensional fBm with Hurst index

$$H = 0.721.$$

Let  $\gamma$  be a random variable of  $\nu$ -dimensional following standard fBm. Set

$$\zeta_\theta : \mathbb{R}^\nu \rightarrow \mathbb{R}^x, \quad \mu_\theta : [0, T] \times \mathbb{R}^x \rightarrow \mathbb{R}^x, \quad \sigma_\theta : [0, T] \times \mathbb{R}^x \rightarrow \mathbb{R}^{x \times w}, \quad \alpha_\theta \in \mathbb{R}^{m \times x}, \quad \beta_\theta \in \mathbb{R}^m,$$

where  $\zeta_\theta, \mu_\theta, \sigma_\theta, \alpha_\theta$ , and  $\beta_\theta$  are all neural networks with a common parameter  $\theta$ . The hyperparameter  $x$  is used to describe the size of the hidden state dimension. Now, define fNSDE as below:

$$\begin{aligned} dX(t) &= \mu_\theta(t, X(t))dt + \sigma_\theta(t, X(t))dB^*(t), \quad t \in (0, T], \\ X(0) &= X_0 = \zeta_\theta(\gamma), \\ Y(t) &= \alpha_\theta X(t) + \beta_\theta, \quad t \in [0, T], \end{aligned} \tag{2.3}$$

where

$$X(t) : [0, T] \rightarrow \mathbb{R}^x$$

is a latent variable such that each generative path of  $Y(t)$  can approximate  $Y_{\text{true}}$  in some probabilistic sense. In fact, (2.3) is an extension of the NSDE generator shown in the paper [5, (3)]. However, this extension is non-trivial because we change the noise driving mechanism from pure Brownian motion to fBm so that (2.3) can generate a time series with long-range dependency, therefore better simulating the target path of CATL's closing stock price.

**Remark 2.1.** *The existence and uniqueness of the latent variable  $X(t)$  can be guaranteed by imposing some mild conditions, such as the global-Lipschitz condition, on drift and diffusion terms for SDE, see; e.g., [4]. Particularly, Hayashi et al. have proven in [4, Theorem V.1] that when*

$$H > \frac{1}{2},$$

*$\mu_\theta$  and  $\sigma_\theta$  have multi-layer perceptron (MLP) network structures with the tanh activation function; the strong solution  $X(t)$  is existence and uniqueness. Moreover, the globally Lipschitz network (drift term and diffusion term) in (2.3) ensures that the price dynamics do not exhibit wild, unbounded behavior. It guarantees that the model behaves smoothly, preventing the model from overfitting to noise or large market swings [27], thus providing more reliable and stable predictions for stock prices.*

In order to further efficiently simulate relationships between widely separated time points, we additionally incorporate the self-attention module into the generator.

**Self-attention.** Let

$$\Delta t = T/n$$

with  $n$  being a positive integer number. Denote by

$$\mathbf{Y} := (Y(t_1), \dots, Y(t_i), \dots, Y(t_n)) \in \mathbb{R}^{m \times n}$$

the value of  $Y(t)$  at time point

$$t_i := i\Delta t, \quad i = 1, \dots, n,$$

where  $Y(t_i) \in \mathbb{R}^m$ . Let  $W_1, W_2 \in \mathbb{R}^{m \times m}$  be the learned weight matrices. Define

$$\phi_1(Y(t_i)) := W_1 Y(t_i), \quad \phi_2(Y(t_i)) := W_2 Y(t_i).$$

Set

$$\tau_{ji} := \frac{\exp(s_{ij})}{\sum_{i=1}^n \exp(s_{ij})}, \quad s_{ij} := \phi_1(Y(t_i))^T \phi_2(Y(t_j)),$$

where  $\tau_{ji}$  stands for the extent to which the model attends to the  $i$ -th time point when generating  $Y(t_j)$  [28]. In other words,  $\tau_{ji}$  represents an attention weight coefficient between  $t_i$  and  $t_j$ ,  $i, j = 1, \dots, n$ . Let

$$\phi_3(Y(t_i)) := W_3 Y(t_i), \quad \phi_4(Y(t_i)) := W_4 Y(t_i)$$

with  $W_3, W_4 \in \mathbb{R}^{m \times m}$  denoting the other two learned weight matrices. Define the output of the attention layer by

$$\mathbf{Z} := (Z(t_1), Z(t_2), \dots, Z(t_i), \dots, Z(t_n)) \in \mathbb{R}^{m \times n}$$

with

$$Z(t_i) := \phi_4\left(\sum_{k=1}^n \tau_{i,k} \phi_3(Y(t_k))\right).$$

Next, we aim to combine this output with  $Y(t_i)$  to increase global relations. Precisely denote the final output by

$$\mathbf{O} = (O(t_1), \dots, O(t_i), \dots, O(t_n))$$

with

$$O(t_i) = \epsilon Z(t_i) + Y(t_i),$$

where  $\epsilon$  is a learnable scalar parameter and it is initialized as 0. It is notable that introducing the learnable  $\epsilon$  allows the network to first rely on the information in the local neighborhood (because this is easier), and then gradually learns to assign more weights of non-local cues.

Define the continuous version of  $\mathbf{O}$  by

$$O(t) : [0, T] \rightarrow \mathbb{R}^m,$$

where the initial value

$$O(0) := O_0 = Y(0).$$

Each sample path of  $O(t)$  is infinite dimensional.  $O(t)$  will be regarded as inputs to the discriminator. We now briefly describe discriminator.

**Discriminator.** Let

$$\xi_\rho : \mathbb{R}^m \rightarrow \mathbb{R}^h, \quad f_\rho : [0, T] \times \mathbb{R}^h \rightarrow \mathbb{R}^h, \quad g_\rho : [0, T] \times \mathbb{R}^h \rightarrow \mathbb{R}^{h \times m}, \quad q_\rho \in \mathbb{R}^h,$$

where  $\xi_\rho, f_\rho, g_\rho, q_\rho$  are all global-Lipschitz neural networks parameterised collectively by  $\rho$ , and the dimension  $h$  is a hyperparameter representing the size of hidden state. The discriminator is expressed as another NSDE of the form: for  $H(t) : [0, T] \rightarrow \mathbb{R}^h$ ,

$$\begin{aligned} dH(t) &= f_\rho(t, H(t))dt + g_\rho(t, H(t))dO(t), \quad t \in (0, T], \\ H(0) &= \xi_\rho(O_0), \\ D &= q_\rho \cdot H(T), \end{aligned} \tag{2.4}$$

where  $g_\rho(t, H(t))dO(t)$  denotes the Riemann-Stieltjes integral related to  $O(t)$ . The value  $D \in \mathbb{R}$  is a function of the terminal hidden state  $H(T)$ . It is the discriminator's score for real versus fake. In order to intuitively see how the extended fNSDE-Net works, we present its flowchart in Figure 3.

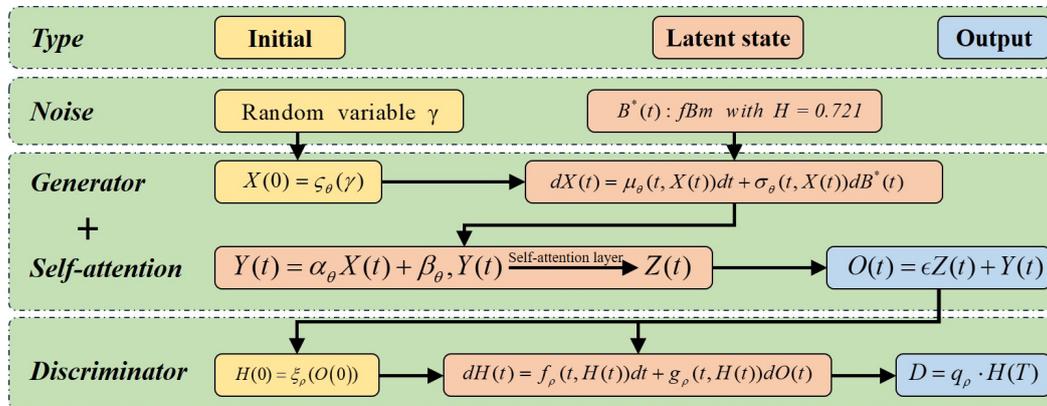


Figure 3. Flowchart of extended fNSDE-Net.

**Remark 2.2.** The well-posedness of SDE shown in (2.4) can be guaranteed since  $f_\rho$  and  $g_\rho$  are global Lipschitz [5]. The construction of the discriminator (2.4) is inspired by the formulation of NCDE shown in [29] with respect to the control  $O(t)$ . Especially, we mention that the discriminator (2.4) is driven by fBm, which is different from the classical discriminator [5, (5)] driven by pure Brownian motion.

With the model specified, we are now in a position to focus on the data training. Suppose we observe an irregularly sampled time series

$$\mathbf{y} = (y_0, \dots, y_i, \dots, y_n)$$

from  $Y_{\text{true}}$ . Denote the linear interpolation of  $\mathbf{y}$  by  $\hat{y}(t)$  such that

$$\hat{y}(t_i) = y_i, \quad i = 0, \dots, n,$$

and define  $\hat{H}(t)$  by following SDE:

$$\begin{aligned} d\hat{H}(t) &= f_\rho(t, \hat{H}(t))dt + g_\rho(t, \hat{H}(t))d\hat{y}(t), \quad t \in (0, T), \\ \hat{H}(0) &= \xi_\rho(\hat{y}(t_0)), \\ \hat{D} &= q_\rho \cdot \hat{H}(T), \end{aligned} \tag{2.5}$$

where  $g_\rho(t, \hat{H}(t))d\hat{y}(t)$  is the Riemann-Stieltjes integral, depending on the regularity of  $\hat{y}(t)$ .

The training loss function applied in this work is the Wasserstein loss, which is a common loss function in GAN, see, e.g., [30, 31]. Let

$$O_\theta : (\gamma, B^*(t)) \mapsto O(t)$$

be the overall action of the generator with self-attention. Let

$$D_\rho : O(t) \mapsto D$$

represent the overall action of the discriminator. Similarly, let

$$\hat{D}_\rho : \hat{y} \mapsto \hat{D}$$

denote the overall action of the system (2.5). Then the generator is optimized by

$$\min_{\theta} \{\mathbb{E}[D_\rho(O_\theta(\gamma, B^*(t)))]\},$$

and the discriminator is optimized by

$$\max_{\rho} \{\mathbb{E}[D_\rho(O_\theta(\gamma, B^*(t)))] - \mathbb{E}[\hat{D}_\rho(\hat{y})]\},$$

where  $\mathbb{E}[\cdot]$  stands for the expectation. Training is performed via stochastic gradient descent [32] techniques as usual.

### 3. Numerical experiments

This section is devoted to comparing the extended fNSDE-Net with several other network models through numerical experiments to highlight the superior performance of the proposed method in terms of fitting and generalization. Before doing this, we first normalize the observed data  $\mathbf{y}$  using the MinMaxScaler provided by the Scikit-learn preprocessing library to avoid excessively large and unstable weight value distributions during the calculation process. Specifically, we define  $\mathbf{y}_{\text{scaled}}$  according to  $\mathbf{y}$  as below:

$$\mathbf{y}_{\text{scaled}} := \frac{\mathbf{y} - \min\{\mathbf{y}\}}{\max\{\mathbf{y}\} - \min\{\mathbf{y}\}}.$$

The MinMax scaler is chosen here because it preserves the shape of the dataset without introducing any distortion, and Tee et al. [33] have employed this technique to normalize the data.

We select the stock closing prices of CATL from May 28, 2023, to May 28, 2024, a total of 242 data points, as observations. Therefore,

$$\mathbf{y}_{\text{scaled}} = (y_{\text{scaled}}(1), \dots, y_{\text{scaled}}(242)).$$

Moreover, considering the timeliness of stocks, we intend to make short-term predictions of stock prices and use the last small part of the dataset as a testing set to demonstrate the generalization performance of the proposed method. Specifically, we take the first  $J_1 := 236$  data for training and the remaining  $J_2 := 6$  data for testing. Define

$$\{\mathbf{x}_{\text{train}}^k(j)\}_{j=1, \dots, J_1} \in \mathbb{R}^{J_1 \times 1}$$

and

$$\{\mathbf{x}_{\text{test}}^k(j)\}_{j=1, \dots, J_2} \in \mathbb{R}^{J_2 \times 1}$$

as the numerical solutions generated by the extended fNSDE-Net corresponding to the  $k$ -th sample of the training set and the testing set, respectively, where  $k = 1, \dots, M$  with  $M$  being the total number

of generated sample paths. Let the sample mean of  $\mathbf{x}_{\text{train}}^k(\cdot)$  and  $\mathbf{x}_{\text{test}}^k(\cdot)$  at the  $j$ -th time point be  $\bar{\mathbf{x}}_{\text{train}}(j)$  and  $\bar{\mathbf{x}}_{\text{test}}(j)$ , where  $\bar{\mathbf{x}}_{\text{train}}(j)$  and  $\bar{\mathbf{x}}_{\text{test}}(j)$  are defined by

$$\bar{\mathbf{x}}_{\text{train}}(j) := \frac{1}{M} \sum_{k=1}^M \mathbf{x}_{\text{train}}^k(j), \quad j = 1, \dots, J_1; \quad \bar{\mathbf{x}}_{\text{test}}(j) := \frac{1}{M} \sum_{k=1}^M \mathbf{x}_{\text{test}}^k(j), \quad j = 1, \dots, J_2.$$

The relative fitting error and generalization error are expressed as follows:

$$e_f := \frac{\left( \sum_{j=1}^{J_1} (y_{\text{scaled}}(j) - \bar{\mathbf{x}}_{\text{train}}(j))^2 \right)^{\frac{1}{2}}}{\left( \sum_{j=1}^{J_1} y_{\text{scaled}}(j)^2 \right)^{\frac{1}{2}}}, \quad e_g := \frac{\left( \sum_{j=1}^{J_2} (y_{\text{scaled}}(236 + j) - \bar{\mathbf{x}}_{\text{test}}(j))^2 \right)^{\frac{1}{2}}}{\left( \sum_{j=1}^{J_2} y_{\text{scaled}}(236 + j)^2 \right)^{\frac{1}{2}}}.$$

A series of numerical experiments are conducted, including ablation studies and comparisons with the classical NSDE-GAN method originally proposed by Kidger et al. [5] to show the effectiveness of our model. The more details on hyperparameter settings and implementation of our method, see the code available at <https://github.com/JHUNAI/EfNSDE>.

Table 2 presents the numerical results of fitting error  $e_f$  and generalization error  $e_g$  for different models.

**Table 2.** Comparison of fitting and generalization errors under different models.

Model name	Sample number $M$	Fitting error	Generalization error
		$e_f$	$e_g$
(1) Extended fNSDE-Net	20	3.9345E-2	1.7802E-2
	50	3.8949E-2	1.7411E-2
	100	3.8620E-2	1.6755E-2
	200	3.8416E-2	1.6681E-2
	500	3.8105E-2	1.6504E-2
(2) fNSDE-GAN	20	4.7400E-2	5.8727E-2
	50	4.7294E-2	5.8364E-2
	100	4.6915E-2	5.8077E-2
	200	4.6428E-2	5.6454E-2
	500	4.6421E-2	5.6288E-2
(3) NSDE-self attention GAN	20	3.5433E-2	3.3715E-2
	50	3.5173E-2	3.2636E-2
	100	3.5060E-2	3.2636E-2
	200	3.5012E-2	3.1751E-2
	500	3.4942E-2	3.0443E-2
(4) NSDE-GAN	20	4.0648E-2	8.9443E-2
	50	3.9968E-2	8.4752E-2
	100	3.9752E-2	8.2404E-2
	200	3.9357E-2	7.7452E-2
	500	3.9167E-2	7.5393E-2

Among these models, (1) represents our proposed model; (2) is the model developed by Hayashi et al. [4], which differs from model (1) in that it does not include a self-attention module; (3) is another model that differs from ours in that it uses standard Brownian motion instead of the fBm applied in our model; and (4) is the pure NSDE-GAN model proposed by Kidger et al. [5]. Based on the numerical results shown in Table 2, we can draw the following conclusions:

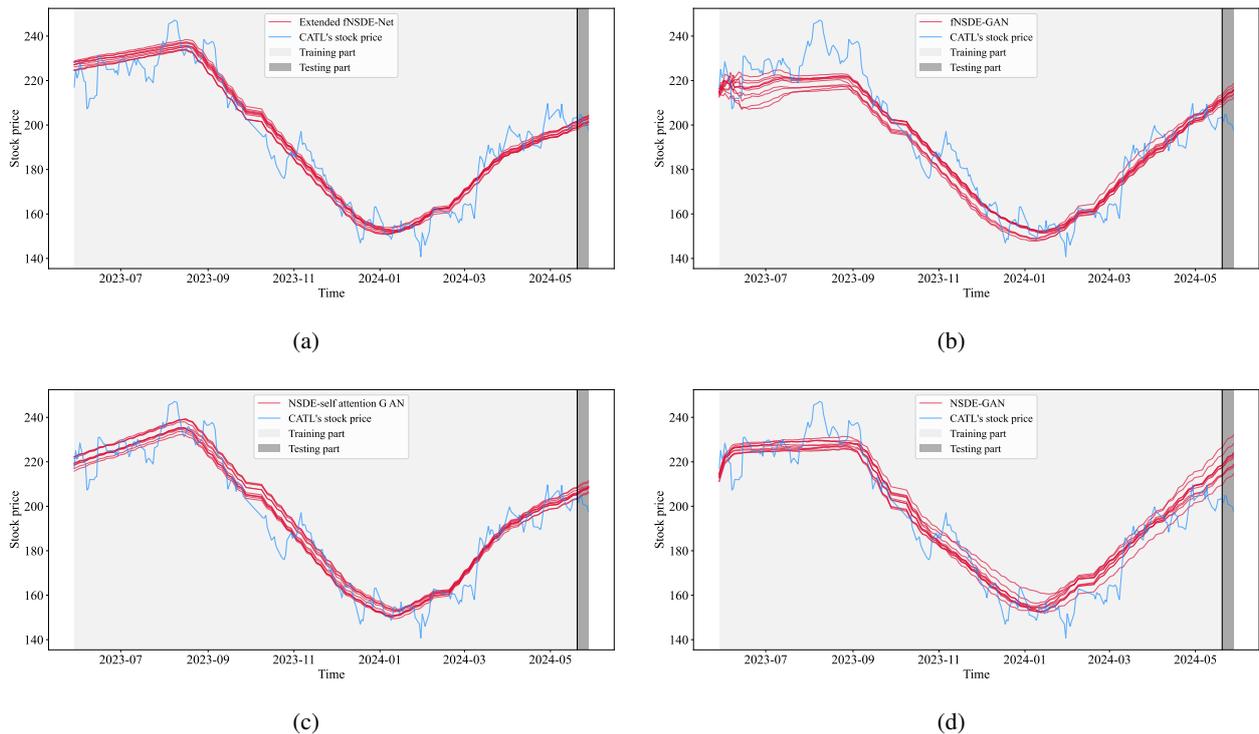
i) Compared with models (2) and (4), our model demonstrates clear advantages in terms of both fitting error and generalization error. Specifically, for a given sample size, the calculation error of our model is smaller.

ii) Compared with model (3), although our model is slightly inferior in terms of fitting error, it shows significant superiority in the calculation of generalization error. In other words, the generalization error computed using model (1) is notably smaller than that obtained from model (3).

Also shown in Figure 4 are 10 sample trajectories generated by different models to intuitively illustrate the fitting and generalization performance of each model. Next, we further investigate the fitting and generalization error performance of different models from the perspective of strong and weak errors. These two errors will be calculated using the last time points of the training and testing sets as examples. The strong and weak errors of fitting and generalization are defined as follows:

$$e_f^{\text{strong}} := \left( \frac{1}{M} \sum_{k=1}^M |y_{\text{scaled}}(J_1) - \mathbf{x}_{\text{train}}^k(J_1)|^2 \right)^{\frac{1}{2}}, \quad e_f^{\text{weak}} := \left| \frac{1}{M} \sum_{k=1}^M (y_{\text{scaled}}(J_1) - \mathbf{x}_{\text{train}}^k(J_1)) \right|,$$

$$e_g^{\text{strong}} := \left( \frac{1}{M} \sum_{k=1}^M |y_{\text{scaled}}(242) - \mathbf{x}_{\text{test}}^k(J_2)|^2 \right)^{\frac{1}{2}}, \quad e_g^{\text{weak}} := \left| \frac{1}{M} \sum_{k=1}^M (y_{\text{scaled}}(242) - \mathbf{x}_{\text{test}}^k(J_2)) \right|.$$



**Figure 4.** Evolution simulation under different models. (a): Extended fNSDE-Net. (b): fNSDE-GAN. (c): NSDE-self attention GAN. (d): NSDE-GAN.

Note that the weak error is related to the approximation of the probability law of the solution, while the strong error measures the deviation from the solution trajectory. Table 3 presents the strong and

weak error behaviors of different models, where models (1)–(4) are consistent with those in Table 2. From Table 3, the following conclusions can be drawn:

i) Compared with models (2) and (4), our model (i.e., model (1)) exhibits smaller fitting and generalization errors in both strong and weak senses, indicating that our model has higher computational accuracy.

ii) Although the strong and weak fitting error calculations of model (3) are slightly better than those of our proposed model (i.e., model (1)), its strong and weak generalization error calculations are significantly worse. Therefore, our proposed model shows a clear overall advantage.

This once again demonstrates the effectiveness of the extended fNSDE-Net.

**Table 3.** Error comparison under different models.

Model name	$M$	Fitting error		Generalization error	
		$e_f^{\text{strong}}$	$e_f^{\text{weak}}$	$e_g^{\text{strong}}$	$e_g^{\text{weak}}$
(1) Extended fNSDE-Net	20	5.2861E-2	4.2061E-2	2.9049E-2	2.4837E-2
	50	5.1636E-2	4.0871E-2	2.6982E-2	2.1333E-2
	100	5.0316E-2	3.9981E-2	2.5961E-2	2.0830E-2
	200	4.9159E-2	3.8864E-2	2.5106E-2	2.0426E-2
	500	4.7349E-2	3.6976E-2	1.9838E-2	1.6932E-2
(2) fNSDE-GAN	20	6.8522E-2	5.5937E-2	8.4207E-2	8.1136E-2
	50	6.1988E-2	4.8794E-2	8.0765E-2	7.7156E-2
	100	6.1498E-2	4.8245E-2	7.6402E-2	7.3110E-2
	200	5.7023E-2	4.4401E-2	6.8641E-2	6.4960E-2
	500	5.6688E-2	4.4183E-2	4.4331E-2	3.8602E-2
(3) NSDE-self attention GAN	20	4.9445E-2	3.8345E-2	5.6734E-2	5.3025E-2
	50	4.5934E-2	3.5982E-2	4.7741E-2	4.3452E-2
	100	4.5888E-2	3.5705E-2	4.5020E-2	4.0167E-2
	200	4.5851E-2	3.5352E-2	3.9213E-2	3.3771E-2
	500	4.5147E-2	3.4774E-2	3.4439E-2	2.7835E-2
(4) NSDE-GAN	20	6.4158E-2	5.0280E-2	1.5875E-1	1.5684E-1
	50	5.9664E-2	4.5697E-2	1.5429E-1	1.5263E-1
	100	5.1682E-2	4.0272E-2	1.1995E-1	1.1781E-1
	200	5.1026E-2	4.0197E-2	7.8652E-2	7.5465E-2
	500	5.0226E-2	3.9292E-2	7.2844E-2	6.9530E-2

#### 4. Bermuda option pricing

This section aims at valuing a vanilla Bermuda call option based on the observed information  $\mathbf{y}$  of CATL's stock price and the simulated paths in the testing set

$$\{\mathbf{x}_{\text{test}}^k(\cdot)\}, \quad k = 1, \dots, M.$$

Note that the Bermuda option is halfway between American and European options, and it can be exercised at a set of prescribed dates within the horizon. The main purpose of this section is to provide an effective way for pricing Bermuda options derived from CATL stock prices. The principal idea is motivated by the LSM method proposed by Longstaff et al. [6], which is a dynamic programming method based on iteratively selecting an optimal policy to estimate a continuation value that determines the optimal exercise strategy. However, unlike [6], our underlying asset prices (i.e., the simulated stock

price paths) are not driven by a geometric Brownian motion (GBM), but are instead computed by a new method proposed in this paper. Next, we intend to use the simulated stock price data in the testing set to value the Bermuda option at the time (denoted as  $t_0$ ) corresponding to the observation value  $y_{j=238}$ , and use this as a numerical example to describe the pricing procedure in detail.

Denote the time stamp corresponding to the observed data  $y_{j=239}, \dots, y_{j=242}$  by  $t_1, \dots, t_4$ . Let the strike price  $K := 200$ , interest rate  $r := 0.5$ . For the sake of clarity, we take 10 sample paths as an example to demonstrate the pricing algorithm. Let  $S_i^j$  be the stock price on the  $i$ -th sample path generated at time  $j$ , where  $i = 1, \dots, 10$ ,  $j = t_1, \dots, t_4$ . Suppose that the exercise waiting time is an equidistant interval

$$\delta t := 0.25.$$

In Table 4, we present the simulated values of CATL stock price from  $t_1$  to  $t_4$  using the extended fNSDE-Net.

**Table 4.** Simulated stock path.

Price Path	Time			
	$t_1$	$t_2$	$t_3$	$t_4$
1	201.88	202.27	202.64	203.03
2	205.73	206.27	206.80	207.29
3	203.71	204.24	204.74	205.25
4	200.48	200.73	201.02	201.36
5	199.54	199.81	200.18	200.52
6	200.85	201.25	201.60	201.94
7	200.90	201.35	201.87	202.33
8	203.06	203.51	203.94	204.38
9	198.98	199.28	199.70	200.04
10	202.23	202.73	203.18	203.69

According to the LSM pricing rule [6], at the final exercise date,  $t_4$ , the option should be exercised if it is in-the-money. However, before the final exercise date, the optimal strategy is to compare the immediate intrinsic value with the expected cash flow from the continuation value, and exercise the option if the intrinsic value is larger.

The cashflow at time  $t_4$  is calculated as  $(K - S_i^{t_4})$ , for  $i = 1, \dots, 10$ , and the relevant results are presented in Table 5.

**Table 5.** Cashflow at  $t_4$ .

Cashflow Path	Time			
	$t_1$	$t_2$	$t_3$	$t_4$
1	–	–	–	3.03
2	–	–	–	7.29
3	–	–	–	5.25
4	–	–	–	1.36
5	–	–	–	0.52
6	–	–	–	1.94
7	–	–	–	2.33
8	–	–	–	4.38
9	–	–	–	0.04
10	–	–	–	3.69

The fitting data for computing intrinsic value and continuation value at time  $t_3$  are shown in Table 6, where  $R_i^j$  represents the discounted cashflow on the  $i$ -th sample path generated at time  $j$ , and the multiplier  $e^{-r\delta t}$  is used as the discount factor.

**Table 6.** Fitting data at  $t_3$ .

Path	$R_i^{t_3}$	$S_i^{t_3}$
1	$3.03 \times e^{-r\delta t}$	202.64
2	$7.29 \times e^{-r\delta t}$	206.80
3	$5.25 \times e^{-r\delta t}$	204.74
4	$1.36 \times e^{-r\delta t}$	201.02
5	$0.52 \times e^{-r\delta t}$	200.18
6	$1.94 \times e^{-r\delta t}$	201.60
7	$2.33 \times e^{-r\delta t}$	201.87
8	$4.38 \times e^{-r\delta t}$	203.94
9	–	–
10	$3.69 \times e^{-r\delta t}$	203.18

The intrinsic value at time  $t_3$  is calculated as  $(K - S_i^{t_3})$ . The continuation value for each sample path at time  $t_3$  is computed using the least squares method with power polynomial basis functions, based on  $S_i^{t_3}$  and the cashflow at  $t_4$ . It is worth noting that Samanez et al. [34] demonstrated that power polynomials provide better approximation performance than other types of basis functions in the LSM method. Additionally, note that the 9-th path does not need to be considered, as it is out-of-the-money (i.e.,  $S_9^{t_3} - K < 0$ ). The calculated data for the continuation value and intrinsic value at  $t_3$  are shown in Table 7, and the cashflow can be updated by comparing these two values, as shown in Table 8.

**Table 7.** Value comparison at  $t_3$ .

Path	Continuation	Intrinsic
1	2.71	2.64
2	6.44	6.80
3	4.61	4.74
4	1.22	1.02
5	0.44	0.18
6	1.76	1.60
7	2.00	1.87
8	3.89	3.94
9	–	–
10	3.20	3.18

**Table 8.** Cashflow:  $t_3$  to  $t_4$ .

Cashflow Path	Time			
	$t_1$	$t_2$	$t_3$	$t_4$
1	–	–	0.00	3.03
2	–	–	6.80	0.00
3	–	–	4.74	0.00
4	–	–	0.00	1.36
5	–	–	0.00	0.52
6	–	–	0.00	1.94
7	–	–	0.00	2.33
8	–	–	3.94	0.00
9	–	–	0.00	0.04
10	–	–	0.00	3.69

Repeating the backward process like  $t_4$  to  $t_3$  and comparing the continuation as well as intrinsic values at time  $t_2$  and  $t_1$  (see Tables 9–13 for intermediate calculation results) allow obtaining the cashflow matrix shown in Table 14 and the optimal stopping time on each sample path; see Table 15. Now, the intrinsic value at time  $t_0$  is computed by

$$(y_{j=238} - K) = 1.56,$$

and the continuation value is calculated via Table 14, i.e., this continuation value is

$$\frac{1}{10}(e^{-4r\delta t} \times (3.03 + 1.36 + 0.52 + 1.94 + 2.33 + 0.04 + 3.69) + e^{-3r\delta t} \times 3.94 + e^{-2r\delta t} \times 4.23 + e^{-r\delta t} \times 5.73) = 1.8889.$$

Hence, the option valuation at time  $t_0$  is

$$\max\{1.56, 1.8889\} = 1.8889.$$

**Table 9.** Fitting data at  $t_2$ .

Path	$R_i^{t_2}$	$S_i^{t_2}$
1	$3.03 \times e^{-2r\delta t}$	202.27
2	$6.80 \times e^{-r\delta t}$	206.27
3	$4.74 \times e^{-r\delta t}$	204.24
4	$1.36 \times e^{-2r\delta t}$	200.73
5	–	–
6	$1.94 \times e^{-2r\delta t}$	201.25
7	$2.33 \times e^{-2r\delta t}$	201.35
8	$3.94 \times e^{-r\delta t}$	203.51
9	–	–
10	$3.69 \times e^{-2r\delta t}$	202.73

**Table 10.** Value comparison at  $t_2$ .

Path	Continuation	Intrinsic
1	2.43	2.27
2	6.00	6.27
3	4.16	4.24
4	1.12	0.73
5	–	–
6	1.56	1.25
7	1.65	1.35
8	3.52	3.51
9	–	–
10	2.83	2.73

**Table 11.** Cashflow:  $t_2$  to  $t_4$ .

Cashflow Path	Time			
	$t_1$	$t_2$	$t_3$	$t_4$
1	–	0.00	0.00	3.03
2	–	6.27	0.00	0.00
3	–	4.23	0.00	0.00
4	–	0.00	0.00	1.36
5	–	0.00	0.00	0.52
6	–	0.00	0.00	1.94
7	–	0.00	0.00	2.33
8	–	0.00	3.94	0.00
9	–	0.00	0.00	0.04
10	–	0.00	0.00	3.69

**Table 12.** Fitting data at  $t_1$ .

Path	$R_i^{t_1}$	$S_i^{t_1}$
1	$3.03 \times e^{-3r\delta t}$	201.88
2	$6.27 \times e^{-r\delta t}$	205.73
3	$4.23 \times e^{-r\delta t}$	203.71
4	$1.36 \times e^{-3r\delta t}$	200.48
5	–	–
6	$1.94 \times e^{-3r\delta t}$	200.85
7	$2.33 \times e^{-3r\delta t}$	200.90
8	$3.94 \times e^{-2r\delta t}$	203.06
9	–	–
10	$3.69 \times e^{-3r\delta t}$	202.23

**Table 13.** Value comparison at  $t_1$ .

Path	Continuation	Intrinsic
1	2.17	1.88
2	5.53	5.73
3	3.72	3.71
4	1.06	0.48
5	–	–
6	1.35	0.85
7	1.39	0.90
8	3.15	3.06
9	–	–
10	2.46	2.23

**Table 14.** Cashflow:  $t_1$  to  $t_4$ .

Path	$t_1$	$t_2$	$t_3$	$t_4$
1	0.00	0.00	0.00	3.03
2	5.73	0.00	0.00	0.00
3	0.00	4.23	0.00	0.00
4	0.00	0.00	0.00	1.36
5	0.00	0.00	0.00	0.52
6	0.00	0.00	0.00	1.94
7	0.00	0.00	0.00	2.33
8	0.00	0.00	3.94	0.00
9	0.00	0.00	0.00	0.04
10	0.00	0.00	0.00	3.69

**Table 15.** Stopping time.

Path	$t_1$	$t_2$	$t_3$	$t_4$
1	0	0	0	✓
2	✓	0	0	0
3	0	✓	0	0
4	0	0	0	✓
5	0	0	0	✓
6	0	0	0	✓
7	0	0	0	✓
8	0	0	✓	0
9	0	0	0	✓
10	0	0	0	✓

In the pricing process described above, we used the LSM pricing framework with extended fNSDE-Net to price a vanilla Bermuda call option. Next, we conduct numerical experiments to compare the reliability of the option price estimates obtained using the LSM with NSDE-GAN against those obtained using our method under different sample sizes. Specifically, we compare the standard deviations (SD) and the lengths of the confidence intervals (CI) for the option price estimates.

Table 16 presents the numerical comparison results, from which it is evident that, for a given sample size, the standard deviation of the estimator calculated using LSM with extended fNSDE-Net (our method) is smaller than that obtained using LSM with NSDE-GAN method. This indicates that our

method is more stable. Furthermore, the length of the confidence interval of the option estimator calculated by our method is shorter than that of the LSM with NSDE-GAN method, suggesting that the estimator obtained by our method is more reliable.

**Table 16.** Comparison of methods with different sample sizes.

Sample numbers (M)	Our proposed method			NSDE-GAN		
	Price	SD	CI length	Price	SD	CI length
20	2.1928	1.8059	1.6904	15.3661	4.8111	4.5033
50	2.6658	1.9921	1.1323	14.1392	4.5948	2.6116
100	3.0678	2.0957	0.8317	14.6104	4.6390	1.8409
200	2.8267	1.9452	0.5425	15.2146	4.7021	1.3113
500	2.8022	1.8484	0.3248	15.1332	4.3830	0.7702
1000	2.7498	1.7992	0.2233	14.8670	4.5773	0.5681
2000	2.7427	1.8617	0.1633	15.1411	4.6163	0.4049
5000	2.7661	1.9128	0.1061	15.0703	4.5138	0.2503
10000	2.7640	1.8487	0.0725	14.9494	4.4896	0.1760

## 5. Conclusions

This paper proposes a new numerical method called extended fNSDE-Net, which is based on GANs with self-attention modules and fNSDE to generate and predict CATL's stock price. This new method not only generates multiple sample paths for the initial input in the sense of probability but also retains the long-term memory characteristics of the generated samples. Numerical experiments show the effectiveness of this new method. Additionally, we calculated the price of Bermuda options derived from stock price paths using the LSM method. The detailed pricing process is demonstrated, and the effectiveness of our pricing method is validated through comparative experiments. Regarding the limitations of this paper, we assume that the drift and diffusion terms of both the generator NSDE and the discriminator NSDE are globally Lipschitz continuous. However, this assumption is made primarily for the convenience of proving the well-posedness of the corresponding NSDE. If we replace the global Lipschitz condition with a weaker, non-global Lipschitz condition, the corresponding NSDE could potentially represent more complex dynamic models. However, this would also complicate the theoretical analysis, which is a topic for our future research.

## Author contributions

Xiao Qi: writing-original draft, data curation, formal analysis, investigation, methodology, validation; Tianyao Duan: software, methodology, data curation; Lihua Wang: conceptualization, supervision, methodology, resources; Huan Guo: conceptualization, formal analysis, investigation, methodology, project administration, resources, supervision, writing-review & editing. All authors have read and agreed to the published version of the manuscript.

## Use of Generative-AI tools declaration

The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article.

## Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grant No. 71601085, the Key Research and Development Project of Hubei Province, China under Grant 2020BCA084, the Research Fund of Jiangnan University under Grant No. 2024JCYJ04, and the Ministry of Education Humanities and Social Sciences Research Project in China under Grant No. 23YJCZH062.

## Conflict of interest

The authors declare no competing interests.

## References

1. H. Fan, Financial analysis in CATL based on Harvard analysis framework, *2022 2nd International Conference on Enterprise Management and Economic Development (ICEMED 2022)*, 2022, 874–879. <https://doi.org/10.2991/aebmr.k.220603.143>
2. H. Liu, H. Liu, H. Wang, Research of power battery risk investment: taking CATL as an example, *Financ. Eng. Risk Manag.*, **6** (2023), 5. <https://doi.org/10.23977/ferm.2023.060501>
3. D. Lyu, Analyzing the risk-return trade-off relationship of Chinese new energy firms using the capital asset pricing model, *High. Bus. Econ. Manag.*, **30** (2024), 429–435. <https://doi.org/10.54097/fwj7mr52>
4. K. Hayashi, K. Nakagawa, Fractional SDE-net: generation of time series data with long-term memory, *2022 IEEE 9th International Conference on Data Science and Advanced Analytics (DSAA)*, 2022. <https://doi.org/10.1109/DSAA54385.2022.10032351>
5. P. Kidger, J. Foster, X. Li, T. J. Lyons, Neural SDEs as infinite-dimensional GANs, *International Conference on Machine Learning*, 2021, 5453–5463. Available form: <https://proceedings.mlr.press/v139/kidger21b.html>.
6. F. A. Longstaff, E. S. Schwartz, Valuing American options by simulation: a simple least-squares approach, *Rev. Financ. Stud.*, **14** (2001), 113–147. <https://doi.org/10.1093/rfs/14.1.113>
7. P. A. Sadorsky, A random forests approach to predicting clean energy stock prices, *J. Risk. Financ. Manag.*, **14** (2021), 48. <https://doi.org/10.3390/jrfm14020048>
8. P. A. Sadorsky, Forecasting solar stock prices using tree-based machine learning classification: How important are silver prices? *North Amer. J. Econ. Finance*, **61** (2022), 101705. <https://doi.org/10.1016/j.najef.2022.101705>
9. M. Wang, Z. Xiao, H. Peng, X. Wang, J. Wang, Stock price prediction for new energy vehicle enterprises: an integrated method based on time series and cloud models, *Expert. Syst. Appl.*, **208** (2022), 118125. <https://doi.org/10.1016/j.eswa.2022.118125>
10. L. Gu, C. Feng, Research on stock price prediction based on Markov-LSTM neural network—take the new energy industry as an example, *Acad. J. Bus. Manag.*, **4** (2022), 42–47. <https://doi.org/10.25236/AJBM.2022.040408>

11. A. Meng, P. Wang, G. Zhai, C. Zeng, S. Chen, X. Yang, et al., Electricity price forecasting with high penetration of renewable energy using attention-based LSTM network trained by crisscross optimization, *Energy*, **254** (2022), 124212. <https://doi.org/10.1016/j.energy.2022.124212>
12. Q. Shen, Y. Zhang, J. Xiao, X. Dong, Z. Lin, Research of daily stock closing price prediction for new energy companies in China, *Data. Sci. Finance Econ.*, **3** (2023), 14–29. <https://doi.org/10.3934/DSFE.2023002>
13. Q. Zhu, X. Zhou, S. Liu, High return and low risk: shaping composite financial investment decision in the new energy stock market, *Energy Econ.*, **122**, (2023), 106683. <https://doi.org/10.1016/j.eneco.2023.106683>
14. G. F. Fan, R. T. Zhang, C. C. Cao, L. L. Peng, Y. H. Yeh, W. C. Hong, The volatility mechanism and intelligent fusion forecast of new energy stock prices, *Financ. Innov.*, **10** (2024), 84. <https://doi.org/10.1186/s40854-024-00621-7>
15. M. M. Alshater, I. Kampouris, H. Marashdeh, O. F. Atayah, H. Banna, Early warning system to predict energy prices: the role of artificial intelligence and machine learning, *Ann. Oper. Res.*, 2022. <https://doi.org/10.1007/s10479-022-04908-9>
16. H. Gao, G. Kou, H. Liang, H. Zhang, X. Chao, C. Li, Y. Dong, Machine learning in business and finance: a literature review and research opportunities, *Financ. Innov.*, **10** (2024), 86. <https://doi.org/10.1186/s40854-024-00629-z>
17. I. Ghosh, R. K. Jana, Clean energy stock price forecasting and response to macroeconomic variables: a novel framework using Facebook’s prophet, neuralprophet and explainable AI, *Technol. Forecast. Soc. Change*, **200** (2024), 123148. <https://doi.org/10.1016/j.techfore.2023.123148>
18. L. Jiang, G. Hu, A review on short-term electricity price forecasting techniques for energy markets, *2018 15th International Conference on Control, Automation, Robotics and Vision (ICARCV)*, 2018, 937–944. <https://doi.org/10.1109/ICARCV.2018.8581312>
19. Y. Lin, Q. Lu, B. Tan, Y. Yu, Forecasting energy prices using a novel hybrid model with variational mode decomposition, *Energy*, **246** (2023), 123366. <https://doi.org/10.1016/j.energy.2022.123366>
20. M. M. Mostafa, A. A. El-Masry, Oil price forecasting using gene expression programming and artificial neural networks, *Econ. Model.*, **54** (2016), 40–53. <https://doi.org/10.1016/j.econmod.2015.12.014>
21. R. Pino, J. Parreno, A. Gomez, P. Priore, Forecasting next-day price of electricity in the Spanish energy market using artificial neural networks, *Eng. Appl. Artif. Intell.*, **21** (2008), 53–62. <https://doi.org/10.1016/j.engappai.2007.02.001>
22. G. J. Lord, C. E. Powell, T. Shardlow, *An introduction to computational stochastic PDEs*, Cambridge University Press, 2014. <https://doi.org/10.1017/CBO9781139017329>
23. F. Biagini, Y. Hu, B. Øksendal, T. Zhang, *Stochastic calculus for fractional Brownian motion and applications*, Springer Science & Business Media Publishers, 2008. <https://doi.org/10.1007/978-1-84628-797-8>

24. X. An, X. Yi, B. Zhou, Q. Zhang, Research on stock market investment model based on time series forecasting and dynamic programming, *Proceedings of the 3rd International Conference on Internet Finance and Digital Economy (ICIFDE 2023)*, 2023, 412–420. [https://doi.org/10.2991/978-94-6463-270-5\\_46](https://doi.org/10.2991/978-94-6463-270-5_46)
25. T. Wang, J. Guo, Y. Shan, Y. Zhang, B. Peng, Z. Wu, A knowledge graph-GCN-community detection integrated model for large-scale stock price prediction, *Appl. Soft. Comput.*, **145** (2023), 110595. <https://doi.org/10.1016/j.asoc.2023.110595>
26. B. Mandelbrot, *Statistical methodology for nonperiodic cycles: from the covariance to R/S analysis*, Nber, 1972, 259–290.
27. M. J. Wainwright, M. I. Jordan, Graphical models, exponential families, and variational inference, *Found. Trends Mach. Learn.*, **1** (2008), 1–305. <https://doi.org/10.1561/22000000001>
28. H. Zhang, I. Goodfellow, D. Metaxas, A. Odena, Self-attention generative adversarial networks, *Proceedings of the 36th International Conference on Machine Learning*, 2019.
29. P. Kidger, J. Morrill, J. Foster, T. Lyons, Neural controlled differential equations for irregular time series, *Adv. Neural Inf. Process. Syst.*, **33** (2020), 6696–6707.
30. M. Arjovsky, S. Chintala, L. Bottou, Wasserstein generative adversarial networks, *ICML'17: Proceedings of the 34th International Conference on Machine Learning*, **70** (2017), 214–223.
31. I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, et al., Generative adversarial nets, *Adv. Neural Inf. Process. Syst.*, **2** (2021), 2672–2680.
32. X. Li, T. K. L. Wong, R. T. Chen, D. K. Duvenaud, Scalable gradients and variational inference for stochastic differential equations, *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, 2020.
33. A. Q. Md, S. Kapoor, A. V. C. Junni, A. K. Sivaraman, K. F. Tee, H. Sabireen, et al., Novel optimization approach for stock price forecasting using multi-layered sequential LSTM, *Appl. Soft Comput.*, **134** (2023), 109830. <https://doi.org/10.1016/j.asoc.2022.109830>
34. U. S. M. de Lima, C. P. Samanez, Complex derivatives valuation: applying the least-squares Monte Carlo simulation method with several polynomial basis, *Financ. Innov.*, **2** (2018), 1. <https://doi.org/10.1186/s40854-015-0019-0>



AIMS Press

©2025 the Author(s), licensee AIMS Press. This is an open access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0>)