Article

Not peer-reviewed version

Efficient Calibration for Option Pricing via a Physics-Informed Chebyshev Kolmogorov-Arnold Network

Sumei Zhang * , Tianci Wu , Haiyang Xiao , Yi Gong * , Weihong Xu

Posted Date: 9 March 2026

doi: 10.20944/preprints202603.0678.v1

Keywords: Kolmogorov-Arnold network; Physical information; stochastic model; differential evolution algorithm; model calibration

Article

Efficient Calibration for Option Pricing via a Physics-Informed Chebyshev Kolmogorov-Arnold Network

Sumei Zhang [1,*], Tianci Wu [2], Haiyang Xiao [2], Yi Gong [3,*] and Weihong Xu [2]

[1] School of Science, Xi'an University of Posts and Telecommunications, Xi'an 71021, China

[2] School of Computer Science and Technology, Xi'an University of Posts and Telecommunications, Xi'an 710121, China

[3] School of Information Communication Engineering, Beijing Information Science and Technology University, Beijing 102206, China

* Correspondence: zhangsumei@xupt.edu.cn (S.Z.); gongyi@bistu.edu.cn (Y.G.)

Abstract

Efficient calibration is essential for the practical application of option pricing models. The Fractional Stochastic Volatility Jump Diffusion (FVSJ) model proposed by Zhang and Yong [1] can reproduce several stylized features observed in option markets, including the volatility smile, volatility clustering, and long-memory effects. However, its multiple stochastic components make conventional calibration computationally expensive. This paper proposes a two-step calibration framework that combines a neural network with a differential evolution (DE) algorithm. In the first step, we construct a Physics-Informed Kolmogorov-Arnold Network (PCKAN) to approximate the FVSJ pricing map. Specifically, we replace the B-spline basis in KAN with second-kind Chebyshev polynomials and incorporate a Black-Scholes PDE residual as an additional penalty term in the training objective, aiming to improve global approximation and enhance numerical stability and interpretability. In the second step, the trained PCKAN is used as a fast surrogate pricer within the DE algorithm to accelerate parameter estimation. Empirical results show that the proposed method achieves calibration accuracy comparable to direct pricing while substantially reducing computational time.

Keywords: Kolmogorov-Arnold network; Physical information; stochastic model; differential evolution algorithm; model calibration

1. Introduction

Model calibration aligns option pricing models with observed market data by minimizing discrepancies between theoretical assumptions and empirical behavior, and is therefore essential for practical derivative valuation as well as for providing quantitative inputs to risk measurement and hedging. Although the Black-Scholes (BS) model [2] remains widely used due to its analytical tractability, it does not adequately reproduce key stylized market features, including volatility smiles, volatility clustering, jumps, and long-memory effects [3–9]. To address these limitations, Zhang, Yong, and Xiao [1] proposed the Fractional Stochastic Volatility Jump Diffusion (FVSJ) model, which combines Heston-type stochastic volatility with fractional volatility dynamics and a mixed exponential jump-diffusion component [11–22]. Notably, Zhang and Sun [10] had previously conducted foundational research on option pricing with double stochastic volatility, stochastic interest rates, and double jumps, providing valuable theoretical for the development of the FVSJ model. However, the increased model complexity and high-dimensional parameterization make calibration in practice computationally intensive and may reduce numerical stability.

Traditional model calibration methods are generally classified into two categories: statistical inference and error minimization. In the statistical inference framework, Lin et al. [23] used maximum likelihood estimation to optimize the free parameters of latent Gaussian models. Mulenga et al. [24]

 doi:10.20944/preprints202603.0678.v1

adopted a Bayesian approach to infer the posterior distribution of time-varying calibrated parameters by combining priors on BS model parameters with observed underlying-asset and option data. In practice, inference-based methods often rely on an analytically tractable likelihood or distributional representation of the model, which can be difficult to obtain for complex settings. In the error-minimization paradigm, Hilliard et al. [25] employed weighted nonlinear least squares to estimate parameters of jump-diffusion models; Bukh et al. [26] proposed an error-minimization algorithm with an adaptive adjustment rate for parameter fitting; and Gao et al. [27] applied a Gauss-Chebyshev approximation for parameter estimation. These approaches are typically formulated as nonconvex optimization problems and may therefore be sensitive to initialization, increasing the risk of convergence to local minima.

Modern deep-learning-based calibration methods are commonly categorized into one-step and two-step approaches. One-step methods learn a direct mapping from market prices to model parameters; for example, Joseph et al. [28] calibrated local volatility and stochastic short-rate models using stock call option and interest rate cap prices, and Shi et al. [29] used SPX option prices as calibration targets. Two-step methods first train a neural network surrogate for the forward mapping from parameters to prices or implied volatilities and then recover parameters via classical optimization or inference; Büchel et al. [30] approximated Trolle-Schwartz prices with an MLP and global optimization, while Dadah [31], Horvath et al. [32], and Gazzani et al. [33] learned mappings from parameters and yield curves to implied volatilities and performed inverse calibration. Compared with one-step methods, two-step methods often provide more stable calibration and retain a clearer structural interpretation of parameters. However, their performance still depends on the surrogate's approximation quality and architecture, and many implementations do not explicitly enforce no-arbitrage conditions, which can lead to economically inconsistent outputs and motivates further methodological refinement.

The Kolmogorov-Arnold Network (KAN) [34] provides an alternative to the multilayer perceptrons commonly used in deep-learning-based calibration and is well suited to function approximation. To further improve its global approximation accuracy, we replace the original B-spline basis functions with second-kind Chebyshev polynomials. To address the lack of explicit financial constraints in many surrogate models, we embed the BS PDE into the loss function, thereby improving interpretability and economic consistency. Building on the two-step calibration paradigm, we integrate the Chebyshev-based KAN with the physical information and propose a Physics-informed Chebyshev KAN network (PCKAN). In implementation, PCKAN is first trained as a surrogate for the FVSJ pricing function, and model parameters are then calibrated via the Differential Evolution algorithm.

The main contributions of this paper are twofold. Firstly, this paper proposes a new physics-informed neural network. Secondly, this paper proposes a new two-step calibration algorithm based on the new physics-informed neural network. The rest of the paper is organized as follows. Section 2 presents the model and calibration problem. Section 3 details calibration algorithm. Section 4 presents some empirical experiments. Section 5 concludes.

2. Model Calibration

2.1. FVSJ Model

Let $\{\Omega, F, \{F_t\}_{0\le t\le T}, \square\}$ denote a complete probability space, where the filtration $\{F_t\}_{0\le t\le T}$ satisfies the right-continuity property, and $\square$ denotes the risk-neutral measure. Within the framework of The Fractional Stochastic Volatility Jump Diffusion model [9], the evolutionary dynamics of the asset price process P_t are characterized by the following system of equations (1):

$$\begin{cases} \dfrac{dP_t}{P_{t-}} = (r-\lambda\delta)dt + \sqrt{v_{1t}}dB_{1t}^P + \sqrt{v_{2t}}dB_{2t}^P + d\left(\sum_{k=1}^{N_t}(\zeta_k - 1)\right), \\ dv_{1t} = k_1(\theta_1 - v_{1t})dt + \sigma_1\sqrt{v_{1t}}dF_{1t}^{H_1}, \\ dv_{2t} = k_2(\theta_2 - v_{2t})dt + \sigma_2\sqrt{v_{2t}}dF_{2t}^{H_2}. \end{cases} \tag{1}$$

where B_{1t}^P and B_{2t}^P are standard Brownian motions, $F_{1t}^{H_1}$ and $F_{2t}^{H_1}$ are fractional Brownian motions with Hurst parameters H_1 and H_2 respectively, and N_t is a Poisson process with intensity λ. Let $\{Z_i\}$ be a sequence of independent and identically distributed non-negative random variables such that $Y = \ln\varsigma$ follows a mixed exponential distribution with probability density function:

$$f_Z(z) = \alpha\sum_{k=1}^{m}\alpha_k\mu_k e^{-\mu_k z}I_{z\geq 0} + \beta\sum_{j=1}^{n}\beta_j\hat{\theta}_j e^{\hat{\theta}_j z}I_{z<0}, \tag{2}$$

where $\mu_k > 1$, $\hat{\theta}_j > 0$, $\alpha \geq 0$, $\beta = 1-\alpha \geq 0$, $\alpha_k, \beta_j \in (-\infty, \infty)$, $\sum_{k=1}^{m}\alpha_k = 1$, $\sum_{j=1}^{n}\beta_j = 1$. To ensure $f_Z(z)$ is a probability density function, a simple sufficient condition is $\sum_{k=1}^{M}\alpha_k\beta_k \geq 0$ $(M = 1,\cdots,m)$, $\sum_{j=1}^{L}\beta_j\hat{\theta}_j \geq 0$ $(L = 1,\cdots,n)$, and the necessary conditions are $\alpha_1 > 0$, $\beta_1 > 0$, $\sum_{k=1}^{m}\alpha_k\eta_k \geq 0$, $\sum_{j=1}^{n}\beta_j\hat{\theta}_j \geq 0$.

According to Thao [36] and Zhang [10], model (1) can be approximated by the following classical stochastic partial integro-differential equation system:

$$\begin{cases} \dfrac{dS_t^{\varepsilon_1,\varepsilon_2}}{S_{t-}^{\varepsilon_1,\varepsilon_2}} = (r-\lambda\delta)dt + \sqrt{v_{1t}}dW_{1t}^S + \sqrt{v_{2t}}dW_{2t}^S + d\left(\sum_{k=1}^{N_t}(\zeta_k - 1)\right), \\ dv_{1t}^{\varepsilon_1} = \left[(H_1 - 1/2)\psi_{1t}\sigma_1\sqrt{v_{1t}^{\varepsilon_1}} + k_1\left(\theta_1 - v_{1t}^{\varepsilon_1}\right)\right]dt + \varepsilon_1^{H_1-1/2}\sigma_1\sqrt{v_{1t}^{\varepsilon_1}}dW_{1t}^v, \\ dv_{2t}^{\varepsilon_2} = \left[(H_2 - 1/2)\psi_{2t}\sigma_2\sqrt{v_{2t}^{\varepsilon_2}} + k_2\left(\theta_2 - v_{2t}^{\varepsilon_2}\right)\right]dt + \varepsilon_2^{H_2-1/2}\sigma_2\sqrt{v_{2t}^{\varepsilon_2}}dW_{2t}^v. \end{cases} \tag{3}$$

Equation (3) is hereafter denoted as the Fractional Stochastic Volatility Jump Diffusion model (the FVSJ model).

Remark: When $\lambda = 0$, $k_2 = 0$, $\theta_2 = 0$, $\sigma_2 = 0$, $H_1 = \dfrac{1}{2}$ and $H_2 = \dfrac{1}{2}$, model (1) reduces to the Heston model.

According to [10], the price of a European put option is approximated as follows:

$$V_T \approx \frac{2}{n-m}e^{-rT}K\sum_{k=0}^{N-1}\Re\left[\phi\left(\frac{k\pi}{n-m}\right)e^{-i\frac{k\pi m}{n-m}}\right]U_k, \tag{4}$$

where

doi:10.20944/preprints202603.0678.v1

$$U_k=\begin{cases}\dfrac{1}{1+\left(k\pi/(n-m)\right)^2}\left[\dfrac{k\pi}{n-m}\sin\left(\dfrac{mk\pi}{n-m}\right)-\cos\left(\dfrac{mk\pi}{n-m}\right)+e^m\right]+\dfrac{m-n}{k\pi}\sin\left(\dfrac{mk\pi}{n-m}\right), & k\neq 0\\ e^m-m-1, & k=0\end{cases},$$

$$\phi(u)=\exp\left\{A(u)+B(u)+C(u)+D(u)\right\},$$

$$A(u)=iu\ln\frac{S_0}{K}+iurT,$$

$$B(u)=\sum_{j=1}^{2}\left\{\frac{(\kappa_j-\gamma_j)\kappa_j\theta_jT}{\Delta_j^{\,2}}-\frac{iu\rho_j\kappa_j\theta_jT}{\Delta_j}+\frac{2\kappa_j\theta_j}{\Delta_j^{\,2}}\ln\frac{2\gamma_j}{2\gamma_j+\left[\kappa_j-\gamma_j-iu\rho_j\Delta_j\right]\left(1-e^{-\gamma_jT}\right)}\right\},$$

$$C(u)=\sum_{j=1}^{2}\frac{iu(iu-1)\left(1-e^{-\gamma_jT}\right)}{2\gamma_j+\left[\kappa_j-\gamma_j-iu\rho_j\Delta_j\right]\left(1-e^{-\gamma_jT}\right)}V_j^{\varepsilon_j},$$

$$D(u)=\lambda T\left(p_u\sum_{i=1}^{m}p_i\eta_i\frac{1}{\eta_i-iu}+q_d\sum_{j=1}^{n}q_j\hat{\theta}_j\frac{1}{\hat{\theta}_j+iu}-1-iu\delta\right),$$

$$\gamma_j=\sqrt{\left[\kappa_j-iu\rho_j\Delta_j\right]^2+iu(1-iu)\Delta_j^{\,2}},$$

$$\Delta_j=\varepsilon_j^{H_j-1/2}\sigma_j,$$

$$\delta=p_u\sum_{i=1}^{m}p_i\eta_i\frac{1}{\eta_i-1}+q_d\sum_{j=1}^{n}q_j\hat{\theta}_j\frac{1}{\hat{\theta}_j+1}-1.$$

K denotes the strike price, T denotes the time to maturity, r denotes the risk-free interest rate. According to [37], the integration interval [c1,c2] is defined as:

$$[m,n]=\left[c_1-L\sqrt{|c_2|},c_1+L\sqrt{|c_2|}\right], \tag{5}$$

where L is a truncation coefficient,

$$c_1=\ln\frac{S_0}{K_0}+\sum_{j=1}^{2}\frac{\alpha_j-U_j}{2k_j}\left(1-e^{-m_j\tau}\right)+\tau\left\{r-\frac{\alpha_1+\alpha_2}{2}+\mu\left[a_u\sum_{i=1}^{m}\frac{a_i}{\xi_i}-b_d\sum_{j=1}^{n}\frac{b_j}{\hat{\alpha}_j}-\left(a_u\sum_{i=1}^{m}a_i\xi_i\frac{1}{\xi_i-1}+b_d\sum_{j=1}^{n}b_j\hat{\alpha}_j\frac{1}{\hat{\alpha}_j+1}-1\right)\right]\right\},$$

$$c_2=\mu\tau\left(a_u\sum_{i=1}^{m}\frac{2a_i}{\xi_i^2}+b_d\sum_{j=1}^{n}\frac{2b_j}{\hat{\alpha}_j^2}\right)+\sum_{j=1}^{2}\left(\Omega_{0j}+\Omega_{1j}e^{-m_j\tau}+\Omega_{2j}e^{-2m_j\tau}\right),$$

 doi:10.20944/preprints202603.0678.v1

$$\Pi_{0j} = \alpha_j\tau + \frac{1}{\mathrm{m}_j}(-\alpha_j + U_j - \alpha_j\Delta_j\rho_j\tau) + \frac{1}{\mathrm{m}_j^2}\left[\Delta_j\rho_j(2\alpha_j - U_j) + \frac{1}{4}\alpha_j\Delta_j^2\tau\right]$$
$$-\frac{\Delta_j^2}{8m_j^3(5\alpha_j - 2U_j)},$$
$$\Pi_{1j} = \frac{\Delta_j\rho_j}{\mathrm{m}_j^2}(U_j - 2\alpha_j) + \frac{1}{\mathrm{m}_j}(1 - \Delta_j\rho_j\tau)(\alpha_j - U_j) + \frac{\alpha_j\Delta_j^2}{2m_j^3} + \frac{\tau\Delta_j^2(\alpha_j - U_j)}{2m_j^2},$$
$$\Pi_{2j} = \frac{\Delta_j^2}{8m_j^3(\alpha_j - 2U_j)},$$
$$\Delta_j = \varepsilon_j^{G_j - 1/2}\sigma_j.$$

2.2. Model Calibration Problem

From an error-minimization perspective, calibration is formulated as an optimization problem. Using observed market prices of European put options together with observable inputs (e.g., strike, risk-free rate, time to maturity, and the underlying price), we calibrate the FVSJ model by estimating its parameter vector. Given a set of model parameters β and option data O, the calibrated parameters are obtained by solving

$$\arg\min \left\| P - FVSJ(O,\beta) \right\|, \tag{6}$$

where $FVSJ(O,\beta)$ is the theoretical option price computed from Equation (4) and P denotes the market price. The objective minimizes the pricing error between them across the available option quotes.

In practice, solving optimization (6) can be computationally expensive because the iterative search requires repeated evaluations of the FVSJ pricing function. To reduce this burden and improve numerical stability, we employ a neural network as a surrogate pricing function and reformulate (6) as an optimization problem based on the network-predicted option price:

$$\arg\min \left\| P - NN(O,\beta) \right\|, \tag{7}$$

where $NN(O,\beta)$ denotes the predicted price produced by the neural network. Through pre-training, the neural network is trained to approximate the theoretical price calculated by equation (4). It can then rapidly process parameter inputs during subsequent calibration, thereby enhancing calibration efficiency.

3. Calibration Algorithm

3.1. PCKAN Neural Network

KAN is a neural network architecture based on the Kolmogorov-Arnold representation theorem. According to [34], most multivariate continuous functions $f\left(x_1, x_2 \cdots x_n\right)$ can be expressed as a combination of a finite number of univariate continuous functions as follows

$$f(x) = f(x_1, \cdots, x_n) = \sum_{q=0}^{2n} \Phi_q\left(\sum_{p=1}^{n} \varphi_{q,p}(x_p)\right), \tag{8}$$

where Φ_q and $\varphi_{q,p}$ are univariate functions. KAN operationalizes this idea by placing learnable univariate functions on network edges: before being aggregated in the next layer, each

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 9 March 2026 doi:10.20944/preprints202603.0678.v1

input channel is transformed by an individual univariate mapping. By contrast, an MLP applies a shared, fixed activation function to its neurons. Accordingly, the layer mappings of KAN and MLP can be written in the simplified forms

$$MLP(x) = (W_L \circ \sigma \circ W_{L-1} \circ \sigma \cdots \circ W_1 \circ \sigma \circ W_0 \circ \sigma), \tag{9}$$

$$KAN(x) = (\Phi_L \circ \Phi_{L-1} \cdots \circ \Phi_1 \circ \Phi_0), \tag{10}$$

where σ denotes the activation function, Φ_i denotes a learnable univariate function. In practice, each univariate function Φ_i is parameterized by a basis expansion, for example,

$$\Phi(x) = \sum_i c_i B_i(x), \tag{11}$$

where c_i are learnable coefficients and $B_i(x)$ are B-spline basis functions defined on a grid.

In option pricing and calibration, inputs are often high-dimensional and large-scale, making the B-spline parameterization in KAN computationally expensive and potentially prone to overfitting. Its local support may also limit the capture of global structure. To address these issues, we replace B-splines with second-kind Chebyshev polynomials, which provide a globally defined orthogonal basis with efficient recurrence relations, stable evaluation, and straightforward tensor-product extensions to higher dimensions. With this modification, we obtain the Chebyshev Kolmogorov-Arnold Network (CKAN). The recurrence relation of the second-kind Chebyshev polynomials is given in equation (12):

$$\begin{aligned} &U_0(x) = 1, \\ &U_2(x) = 2x, \\ &\cdots \\ &U_{n+1}(x) = 2xU_n(x) - U_{n-1}(x) \qquad (n \geq 1). \end{aligned} \tag{12}$$

CKAN retains the same overall architecture as KAN, but replaces the basis representation used within each univariate edge function. A two-layer CKAN is illustrated in Figure 1.

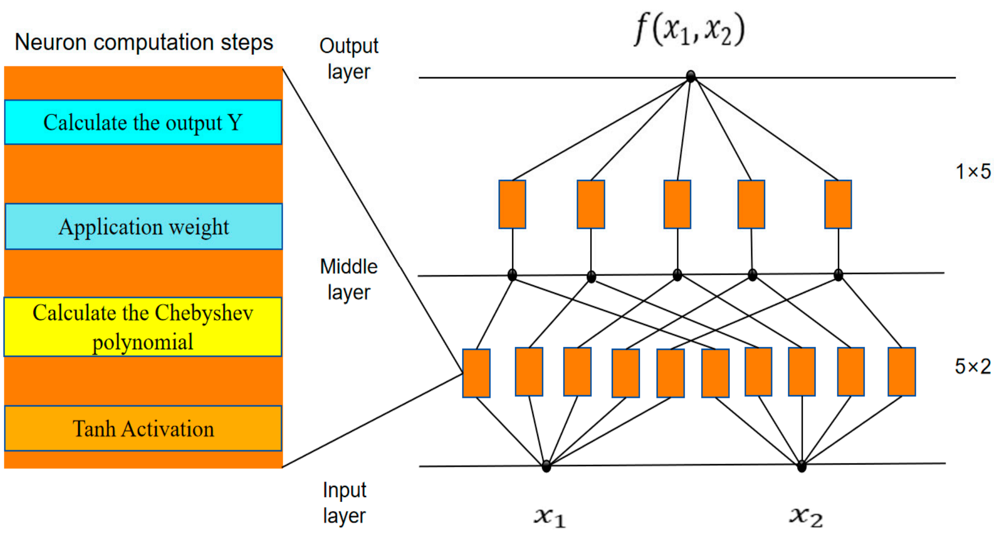

Figure 1. CKAN neural network structure diagram.

 doi:10.20944/preprints202603.0678.v1

As illustrated in Figure 1, CKAN applies a $\mathrm{Tanh}(\cdot)$ normalization to map inputs to [−1,1], expands them using second-kind Chebyshev polynomials, and then aggregates the resulting features through learnable weights. Specifically, the computation follows equations (13)-(17):

$$X_{\mathrm{tanh}} = \mathrm{Tanh}(X), \tag{13}$$

$$\begin{aligned} &U_0 = 1, \\ &U_2 = 2X_{\mathrm{Tanh}}, \\ &\dots \\ &U_{n+1} = 2X_{\mathrm{Tanh}}U_n - U_{n-1} \qquad (n \geq 1), \end{aligned} \tag{14}$$

$$C = stack\left([U_0, U_1 \cdots, U_n]\right), \tag{15}$$

$$S = \sum_{n=0}^{D_l} C_n \widehat{W}_n, \tag{16}$$

$$Y = \sum S \cdot W. \tag{17}$$

These steps can be compactly written as follows

$$Y = \left(\sum_{p=1}^{n-1} W_{q,p} \cdot \left(\sum_{n=0}^{D_l} \widehat{W}_{q,p,n} \cdot U_n(X_p) \right) \right). \tag{18}$$

Equation (18) shows that CKAN employs two sets of parameters: $W_{q,p}$ for linear feature mixing and $W_{q,p}$ for weighting the polynomial expansion. This decoupling provides separate control over linear and nonlinear components, improving modeling flexibility.

Although CKAN improves the basis representation in KAN, it remains data-driven and does not explicitly enforce option-pricing structure (e.g., no-arbitrage and pricing dynamics), which limits financial interpretability. We therefore incorporate the BS PDE [2] as a physics-informed constraint. The BS PDE characterizes the price dynamics with respect to the underlying asset price and time to maturity and provides a standard theoretical foundation for option pricing. Specifically, we embed the BS PDE into the CKAN loss function to obtain the proposed PCKAN model. The BS PDE is given in equation (19).

$$\frac{\partial P}{\partial t} + rS\frac{\partial P}{\partial S} + \tfrac{1}{2}\sigma^2 S^2 \frac{\partial^2 P}{\partial S^2} - rP = 0, \tag{19}$$

where P denotes the option price, t denotes the time to maturity, r is the risk-free interest rate, S denotes the underlying asset price, and σ denotes the volatility of the underlying asset price.

The loss function L_{phys} of PCKAN is defined in equation (20):

$$L_{phys} = \| P_{pred} - P_{\mathrm{real}} \|^2 + \lambda \| rP_{bs} + \frac{\partial P_{bs}}{\partial t} - rS\frac{\partial P_{bs}}{\partial S} - \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 P_{bs}}{\partial^2 S} \|^2 . \tag{20}$$

The PCKAN loss is defined in (20) as the sum of a data-misfit term and a physics-informed term. The first term measures the discrepancy between the predicted option price P_{pred} and the observed price P_{real}, while the second penalizes the BS PDE residual. The hyperparameter λ controls the

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 9 March 2026 doi:10.20944/preprints202603.0678.v1

trade-off between fitting market data and enforcing the PDE constraint. The data flow of PCKAN is shown in Figure 2.

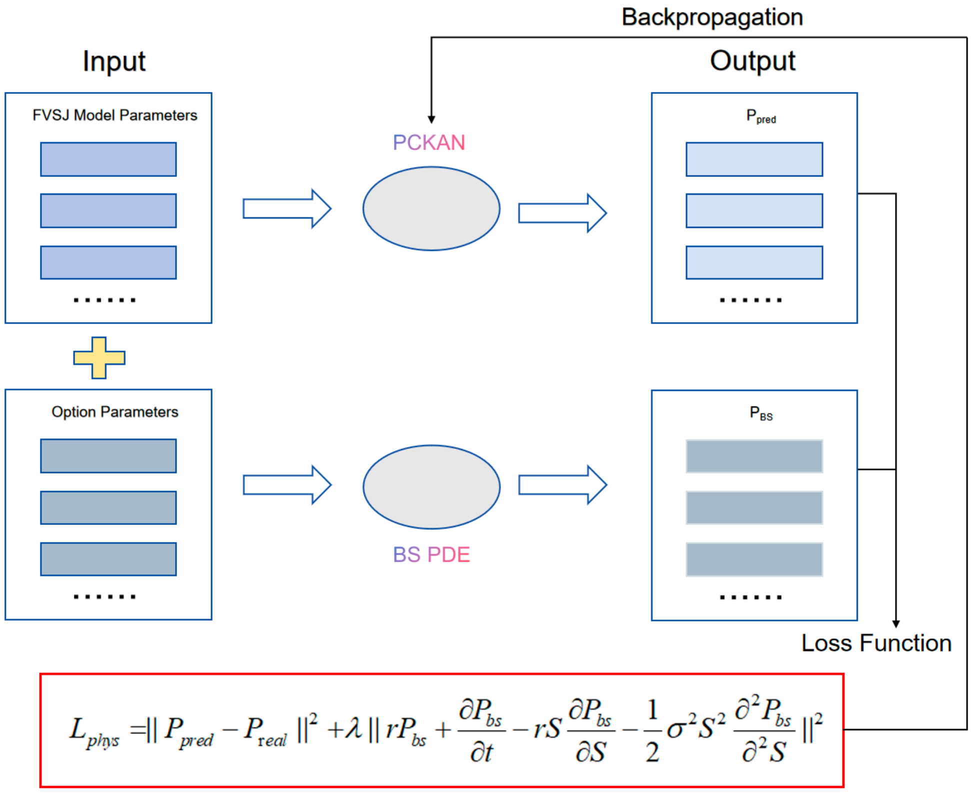

Figure 2. PCKAN data flow diagram.

During training, the loss combines a data-misfit term (between the network-predicted and observed option prices) with a physics term that penalizes the BS PDE residual. The PDE residual is computed via automatic differentiation and evaluated alongside the forward pass, enabling parallel execution and reducing the additional overhead introduced by the physics constraint.

3.2. A Two-Step Calibration Algorithm

We propose a two-step calibration procedure that integrates PCKAN with the DE algorithm. In Step one, PCKAN is trained to approximate the mapping from model parameters and option inputs to option prices. In Step two, the trained network serves as a fast surrogate pricer within DE to identify parameter values that best match market quotes. The overall workflow is summarized in Figure 3.

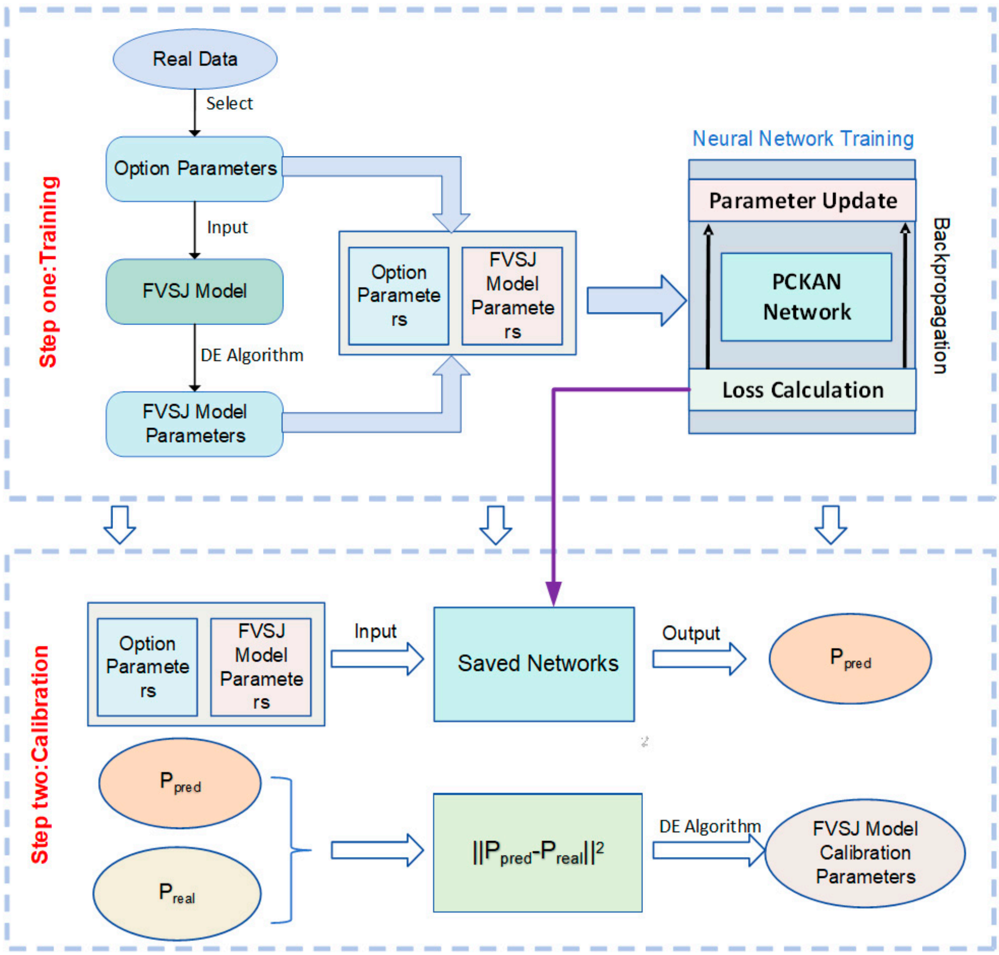

Figure 3. Two-step calibration flowchart.

Because real-market data do not provide ground-truth parameters, calibration is performed by minimizing pricing errors: an initial parameter estimate is obtained via DE and subsequently refined through further optimization. Algorithm 1 details the proposed two-step procedure for the real-data setting.

Algorithm 1. Two-step calibration via PCKAN and DE.

Input: Observed option quotes $\left\{\left(O_i^r, P_i^r\right)\right\}_{i=1}^N$; stochastic pricing model $\mathcal{M}(\cdot)$; discrepancy loss $L(\cdot,\cdot)$; physics-informed loss $L_{phys}(\cdot,\cdot)$; DE setting.

Output: Trained surrogate f_W; calibrated model parameters Ω^c for new quotes.

Step 1 Offline label generation and PCKAN training

Observed option quotes $\left\{\left(O_i^r, P_i^r\right)\right\}_{i=1}^N$; stochastic pricing model $\mathcal{M}(\cdot)$;

discrepancy loss $L(\cdot,\cdot)$; physics-informed loss $L_{phys}(\cdot,\cdot)$; DE setting.

Extract option inputs O_i^r and corresponding market prices P_i^r $(i=1,\cdots,N)$

 doi:10.20944/preprints202603.0678.v1

For $i = 1, \cdots, N$ do

Estimate a parameter label by solving $\Omega_i^r = \arg\min_{\Omega} L\left(\mathcal{M}(O_i^r, \Omega), P_i^r\right)$,

where the minimization is performed using DE.

Compute the model-based target price $P_i^m = \mathcal{M}(O_i^r, \Omega)$

End For

Form the training set $\mathcal{D} = \left\{\left((O_i^r, \Omega_i^r), P_i^m\right)\right\}_{i=1}^{N}$

Train the PCKAN surrogate f_W by minimizing the physics-informed objective

Step 2 Online calibration using the trained surrogate

For a new option input data O_j^r, with its market price P_j^r calibrate

parameters by solving

$$\Omega_j^c = \arg\min_{\Omega} L_{phys}\left(f_{W^*}\left(O_j^r, \Omega\right), P_j^r\right)$$

where the minimization is performed using DE.

Return Ω_j^c.

4. Empirical Analysis

4.1. Data and Preprocessing

To evaluate the applicability and robustness of PCKAN in practice, we conduct an empirical study using China 50ETF option data from Yahoo Finance. To ensure data quality and consistency with the model inputs, we apply the following screening rules:

1.Remove option contracts that violate basic no-arbitrage principles, ensuring equation (21) is satisfied:

$$C \geq \max(0, S-K, S-D-Ke^{-r\tau}), \tag{21}$$

where C denotes the call price and D is the present value of dividends during the option's life.

2.Remove options with less than 1 day to expiry to avoid interference from accelerated time value decay near expiration and also remove options with more than 255 days to expiry as they are less liquid and have higher premiums.

3.Remove option quotes priced below 0.01, as these quotes are usually within the bid-ask spread and have larger price fluctuations.

4.Remove all option data from half-day trading sessions to ensure complete matching with the underlying asset index's trading hours.

After screening, the sample contains 38,610 observations, including variables such as option ID, time, contract type, day to expire, spot Price, and settle Price. For preprocessing, settle Price is standardized using z-score normalization. The data are randomly split into a training set (70%) and a test set (30%). Descriptive statistics are reported in Table 1.

Table 1. Descriptive statistics of the empirical dataset.

Dataset	Basic Data	Mean	Variance	Maximum	Minimum
Training	Day to Expire	87.13	65.70	244.0	1.0
	Spot Price	2.57	0.16	3.0	2.3
	Settle Price	0.18	0.19	0.98	0.01
Test	Day to Expire	95.98	65.14	244.0	1.0
	Spot Price	2.87	0.13	3.1	2.6
	Settle Price	0.17	0.16	0.97	0.01

4.2. Evaluation Metrics and Experimental Design

To evaluate PCKAN for model calibration, we implement three surrogate models in PyTorch, using PCKAN as the proposed method and the multilayer perceptron MLP and KAN as baseline models, and conduct comparative calibration experiments. To ensure comparability, the three networks use closely matched architectures with seven input features, nine outputs, one input layer, five hidden layers, and one output layer; all hidden layers have width 16, and the MLP uses the Softplus activation function. All models are trained for 200 epochs using the Adam optimizer with a stepwise learning-rate schedule. PCKAN is trained with the physics-informed loss in Equation (20), whereas MLP and KAN are trained with the mean squared error (MSE) loss in Equation (22)

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2, \tag{22}$$

where y_i and $\hat{y}_i$ denote the true and predicted values for the i-th sample, and n is the number of samples.

Calibration performance is evaluated using approximation error (AE) and calibration error (CE), defined in Equations (23)-(24)

$$\mathrm{AE} = \frac{1}{n}\sum_{i=1}^{n}(P_i^{True} - P_i^{PCKAN})^2. \tag{23}$$

$$\mathrm{CE} = \frac{1}{n}\sum_{i=1}^{n}(P_i^{True} - FVSJ_i\left(O,\widehat{\beta}\right))^2. \tag{24}$$

where P_i^{TRUE} is the market price, P_i^{PCKAN} is the PCKAN output, and $\widehat{\beta}$ denotes the calibrated parameter vector. All experiments are run on a laptop with an Intel Core i7-14650HX CPU (2.20 GHz) and an NVIDIA GeForce RTX 4060 GPU with 8 GB VRAM.

4.3. Experimental Results and Analysis

Table 2 reports the training mean squared error (MSE) of MLP, KAN, and PCKAN on the empirical Heston and FVSJ datasets.

Table 2. Training MSE of MLP, KAN, and PCKAN on the Heston and FVSJ datasets.

Dataset	MLP	KAN	PCKAN
Heston	2.2×10-4	7.4×10-2	1.6×10-4
FVSJ	5.8×10-4	9.0×10-1	3.2×10-4

Table 2 shows that PCKAN attains the lowest training MSE in both cases, with 1.6×10-4, on the Heston dataset and 3.2×10-4 on the FVSJ dataset, compared with 2.2×10-4 and 5.8×10-4 for MLP and 7.4×10-2 and 9.0×10-1 for KAN, respectively. Overall, these results suggest that PCKAN fits the training data more closely than MLP and KAN under the same experimental setting.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 9 March 2026 doi:10.20944/preprints202603.0678.v1

Calibration is performed on the validation sets of the empirical Heston and FVSJ datasets using the pre-trained PCKAN and MLP surrogates, and the calibration time is recorded. Table 3 reports the resulting calibration errors. Table 3 shows that PCKAN yields lower errors than MLP for both models, with 5.6×10-5 versus 7.4×10-5 on the Heston dataset and 4.0×10-6 versus 8.2×10-6 on the FVSJ dataset. These results indicate that, under the same calibration procedure, PCKAN achieves smaller residual pricing errors on the validation data than the MLP benchmark.

Table 3. Comparison of calibration errors for PCKAN and MLP on the Heston and FVSJ datasets.

Dataset	MLP	PCKAN
Heston	7.4×10-5	5.6×10-5
FVSJ	8.2×10-6	4.0×10-6

Furthermore, Figures 4–7 compare the fitted option prices produced by MLP and PCKAN against market prices for the Heston and FVSJ datasets.

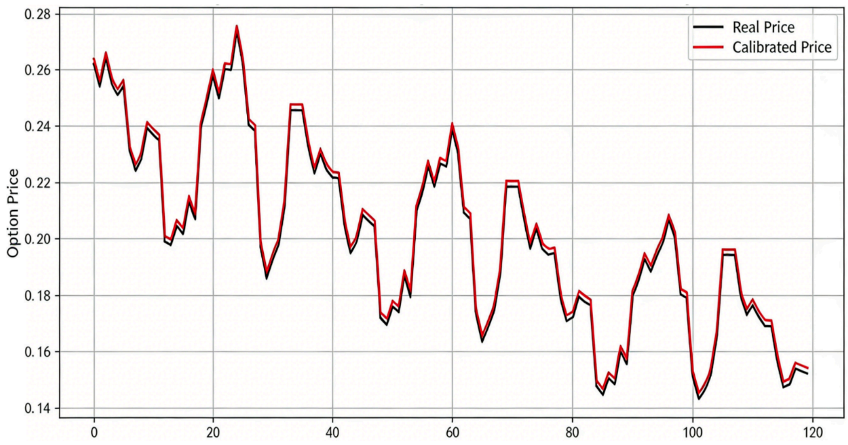

Figure 4. PCKAN: Predicted vs. market option prices on the FVSJ dataset.

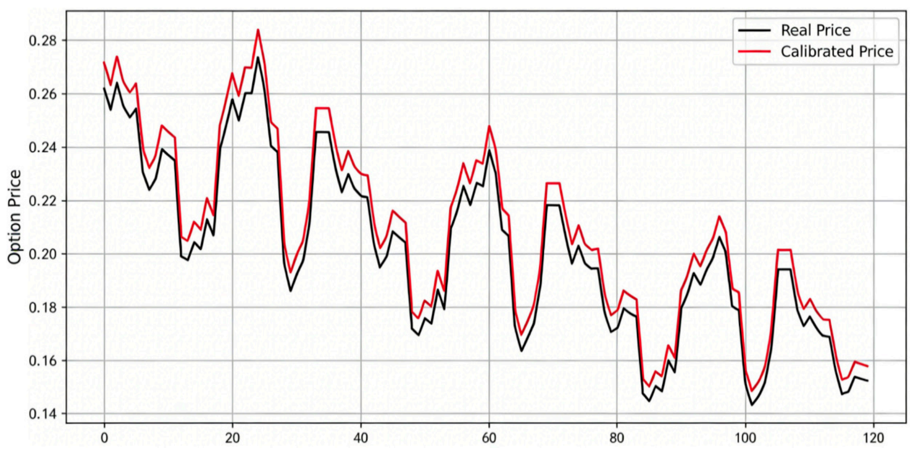

Figure 5. PCKAN: Predicted vs. market option prices on the Heston dataset.

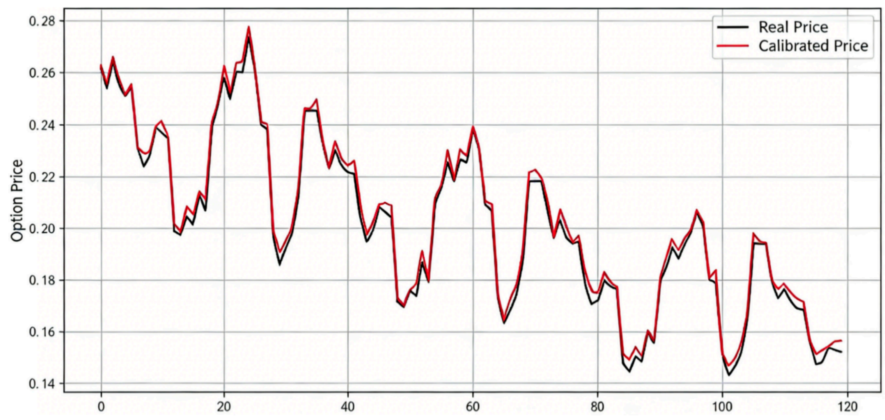

Figure 6. MLP: Predicted vs. market option prices on the FVSJ dataset.

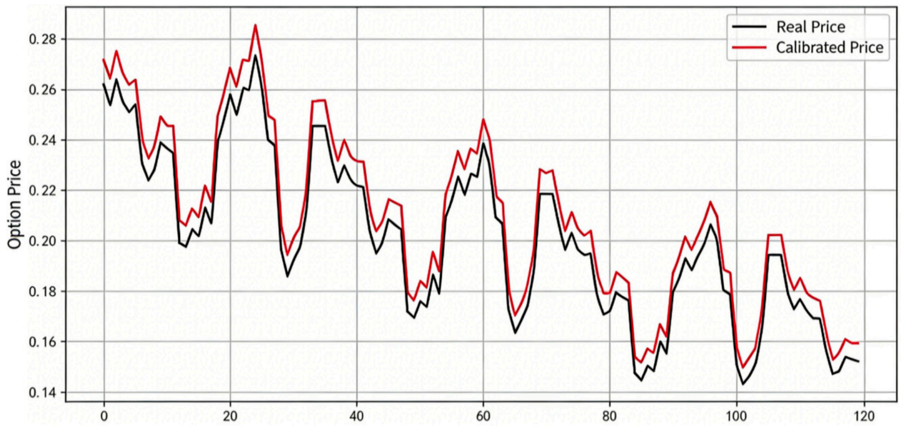

Figure 7. MLP: Predicted vs. market option prices on the Heston dataset.

Figures 4–7 show that across both model settings, PCKAN tracks the market prices more closely than MLP, with smaller deviations in periods of pronounced market stress, such as the 2018 drawdown and the heightened volatility during the 2020 pandemic. Notably, PCKAN achieves higher calibration accuracy when paired with the FVSJ model than with the Heston model. A plausible explanation is that the jump and stochastic-variance components in FVSJ introduce stronger nonlinearities, for which the physics-informed training objective provides a more effective regularization signal, improving fit in regions associated with jumps and tail behavior.

Table 4 reports the average calibration time per observation. For the Heston dataset, PCKAN requires 7 s per calibration, compared with 3 s for MLP and 34 s for the benchmark method. For the FVSJ dataset, the corresponding times are 37 s, 17 s, and 740 s, respectively. Although PCKAN is slower than MLP, it substantially reduces runtime relative to the benchmark method, by factors of 4.9 (Heston) and 20.0 (FVSJ).

Table 4. Calibration time per observation for PCKAN, MLP and benchmark method.

Dataset	MLP	PCKAN	Benchmark Method
Heston	7	3	34
FVSJ	37	17	740

To illustrate the implications at scale, calibrating the full FVSJ sample would take approximately 40 hours with PCKAN versus about 330 hours with the benchmark method, representing an 8.25-fold reduction in total wall-clock time. These results suggest that the benchmark method can be computationally prohibitive for large datasets, whereas PCKAN offers a more practical runtime for batch calibration.

doi:10.20944/preprints202603.0678.v1

5. Conclusions

This study proposes a two-step calibration framework based on a Physics-Informed Chebyshev Kolmogorov-Arnold Network (PCKAN) to improve the efficiency-accuracy balance in calibrating the FVSJ model. The framework replaces the B-spline basis in KAN with second-kind Chebyshev polynomials to reduce computational overhead and incorporates Black-Scholes partial differential equation constraints to impose pricing structure. PCKAN is trained offline as a surrogate for the FVSJ pricer and then used online for fast parameter calibration to market quotes.

Empirical experiments are used to evaluate the method. Across the Heston and FVSJ settings, PCKAN yields lower training and calibration errors than the multilayer perceptron and the original KAN under the same experimental configuration, while substantially reducing runtime relative to benchmark calibration based on repeated model pricing. The empirical results further show that the calibrated prices produced by PCKAN track observed market prices across different market regimes.

Author Contributions: Conceptualization, Sumei Zhang; Formal analysis, Sumei Zhang and Tianci Wu; Methodology, Yi Gong; Validation, Sumei Zhang and Haiyang Xiao; Writing original draft, Tianci Wu; Writing review & editing, Haiyang Xiao and Weihong Xu.

Funding: This work is funded by the National Natural Science Foundation (No. 12401502) and the Natural Science Foundation of Shaanxi Province (No. 2024JC-YBMS-008).

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Zhang S, Yong H, Xiao H. Option pricing with fractional stochastic volatilities and jumps. Fractal and Fractional, 2023; 7(9): 680.
2. Black F, Scholes M. The pricing of options and corporate liabilities. Journal of Political Economy, 1973; 81(3): 637-654.
3. Hoang T M. Long-horizon asset and portfolio returns revisited: Evidence from US markets. Cogent Economics & Finance, 2023; 10(2): 1-16.
4. De Domenico F, Livan G, Montagna G, et al. Modeling and simulation of financial returns under non-Gaussian distributions. Physica A: Statistical Mechanics and its Applications, 2023; 622: 128886.
5. Chong C H, Todorov V. Volatility of volatility and leverage effect from options. Journal of Econometrics, 2024; 240(1): 105669.
6. Alraddadi R. The logTG-SV model: A threshold-based volatility framework with logarithmic shocks for exchange rate dynamics. AIMS Mathematics, 2025; 10: 19495-19511.
7. Kalovwe S K, Mwaniki J I, Simwa R O. On regime-switching European option pricing. Cogent Economics & Finance, 2023; 11(1): 2203439.
8. Singh A, Jha A K, Kumar A N. Prediction of cryptocurrency prices through a path dependent Monte Carlo simulation. Communications in Statistics-Simulation and Computation, 2025: 1-20.
9. Yue C, Shen C. Fractal barrier option pricing under sub-mixed fractional Brownian motion with jump processes. AIMS Mathematics, 2024; 9(11): 31010-31029.
10. Zhang S, Sun Y. Forward starting options pricing with double stochastic volatility, stochastic interest rates and double jumps. Journal of Computational and Applied Mathematics, 2017: 34-41.
11. Merton R C. Option pricing when underlying stock returns are discontinuous. Journal of Financial Economics, 1976; 3: 125-144.
12. Kou S G. A jump diffusion model for option pricing. Management Science, 2002; 48: 1086-1101.
13. Cai N, Kou S G. Pricing Asian options under a hyper-exponential jump diffusion model. Operations Research, 2012; 60: 64-77.
14. Cai N, Kou S G. Option pricing under a mixed-exponential jump diffusion model. Management Science, 2011; 57: 2067-2081.
15. Gatheral J, Jaisson T, Rosenbaum M. Volatility is rough. In: Commodities. Boca Raton: Chapman and Hall/CRC; 2022: 659-690.

 doi:10.20944/preprints202603.0678.v1

16. Fukasawa M. Volatility has to be rough. Quantitative Finance, 2021; 21: 1-8.
17. Cox J C, Ross S A. The valuation of options for alternative stochastic processes. Journal of Financial Economics, 1976; 3: 145-166.
18. White A D, Hull J C. The pricing of options on assets with stochastic volatilities. The Journal of Finance, 1987; 42: 281-300.
19. Heston S L. A closed-form solution for options with stochastic volatility with applications to bond and currency options. Review of Financial Studies, 1993; 6: 327-343.
20. Wang X, Xiao W, Yu J. Modeling and forecasting realized volatility with the fractional Ornstein Uhlenbeck process. Journal of Econometrics, 2023; 232(2): 389-415.
21. Zhang S, Yong H, Xiao H. Option pricing with fractional stochastic volatilities and jumps. Fractal and Fractional, 2023; 7(9): 680.
22. Christoffersen P, Heston S, Jacobs K. The shape and term structure of the index option smirk: Why multifactor stochastic volatility models work so well. Management Science, 2009; 55: 1914-1932.
23. Lin A, Tolooshams B, Atchadé Y, et al. Probabilistic unrolling: Scalable, inverse-free maximum likelihood estimation for latent Gaussian models. In: Proceedings of the International Conference on Machine Learning, PMLR, 2023; 21153-21181.
24. Mulenga N M, Fu Y. Dynamic calibration based on the Black-Scholes option pricing model by Bayesian method. IEEE Access, 2024; 12: 119314-119326.
25. Hilliard J E, Hilliard J, Ngo J T D. Implied parameter estimation for jump diffusion option pricing models: Pricing accuracy and the role of loss and evaluation functions. Journal of Commodity Markets, 2024; 35: 100408.
26. Bukh A V, Kashtanova S V, Shepelev I A. Complex error minimization algorithm with adaptive change rate. Chaos, Solitons & Fractals, 2023; 176: 114154.
27. Gao J, Jia R, Noorani I, et al. Calibration of European option pricing model in uncertain environment: Valuation of uncertainty implied volatility. Journal of Computational and Applied Mathematics, 2024; 447: 115890.
28. Joseph B, Loeper G, Obłój J. Joint calibration of local volatility models with stochastic interest rates using semimartingale optimal transport. Quantitative Finance, 2024; 24(11): 1597-1620.
29. Shi Y, Teng B, Wang S. Option pricing mechanisms driven by backward stochastic differential equations. Financial Innovation. 2025;11(1):90.
30. Büchel P, Kratochwil M, Nagl M, et al. Deep calibration of financial models: turning theory into practice. Review of Derivatives Research, 2022; 25(2): 109-136.
31. Dadah S. Deep calibration of option pricing models. Cape Town: University of Cape Town, 2022.
32. Horvath B, Muguruza A, Tomas M. Deep learning volatility: a deep neural network perspective on pricing and calibration in (rough) volatility models. Quantitative Finance, 2021; 21(1): 11-27.
33. Gazzani G, Guyon J. Pricing and calibration in the 4-factor path-dependent volatility model. Quantitative Finance. 2025;25(3):471-489.
34. Somvanshi S, Javed S A, Islam M M, et al. A survey on kolmogorov-arnold network. ACM Computing Surveys, 2025; 58(2): 1-35.
35. Bae H O, Kang S, Lee M, et al. Option pricing and local volatility surface by physics-informed neural network. Computational Economics, 2024; 64(5): 3143-3159.
36. Thao T H. An approximate approach to fractional analysis for finance. Nonlinear Analysis: Real World Applications, 2006; 7(1): 124-132.
37. Fang F, Oosterlee C W. A novel pricing method for European options based on Fourier-cosine series expansions. SIAM Journal on Scientific Computing, 2008; 31: 826-848.