Neural option pricing for rough Bergomi model

TENG Changqing, LI Guanglian
u3553440@connect.hku.hk, lotusli@hku.hk

香 港 大 學
THE UNIVERSITY OF HONG KONG

10th International Congress on Industrial and Applied Mathematics
ICIAM 2023 TOKYO

Abstract

The introduction of the fractional kernel makes the rough Bergomi model have a remarkable fit to the implied volatilities, but also imposes challenges to simulation and option pricing. We propose an efficient simulation scheme for the fractional kernel by sum-of-exponentials (SOE) on the interval $[\tau, T]$ and utilize the exact method near the singularity. The initial forward variance curve is treated as an unknown function and approximated by a neural network using the samples generated via the SOE scheme. We implement Wasserstein-1 distance as the loss function. The experiment results indicate that the error of the option price can be bounded by the very Wasserstein distance attained during training.

Introduction

The **rough Bergomi model** is given by the system

$$S_t = S_0 \exp\left(-\frac{1}{2}\int_0^t V_s \mathrm{d}s + \int_0^t \sqrt{V_s}\left(\rho \mathrm{d}W_s + \sqrt{1-\rho^2}\mathrm{d}W_s^{\perp}\right)\right)$$

$$V_t = \xi_0(t)\exp\left(\eta\sqrt{2H}\int_0^t (t-s)^{H-\frac{1}{2}}\mathrm{d}W_s - \frac{\eta^2}{2}t^{2H}\right) \quad t \in [0,T]$$

- **fractional kernel**: $G(t) = t^{H-\frac{1}{2}}$
- **stochastic Volterra process**: $I(t) = \sqrt{2H}\int_0^t (t-s)^{H-\frac{1}{2}}\mathrm{d}W_s$
- **stock process**: S_t with initial stock price $S_0 = 1$
- **volatility process**: V_t with initial forward variance curve $\xi_0(t) = \mathbb{E}[V_t]$
- **parameters**

- $H \approx 0.1$: Hurst index, controls the decay of term structure of skew for small T
- ρ: correlation between the Brownian motion driving V and S, $\rho \in (-1,0)$
- η: the product $\rho\eta$ controls the ATM skew for large T

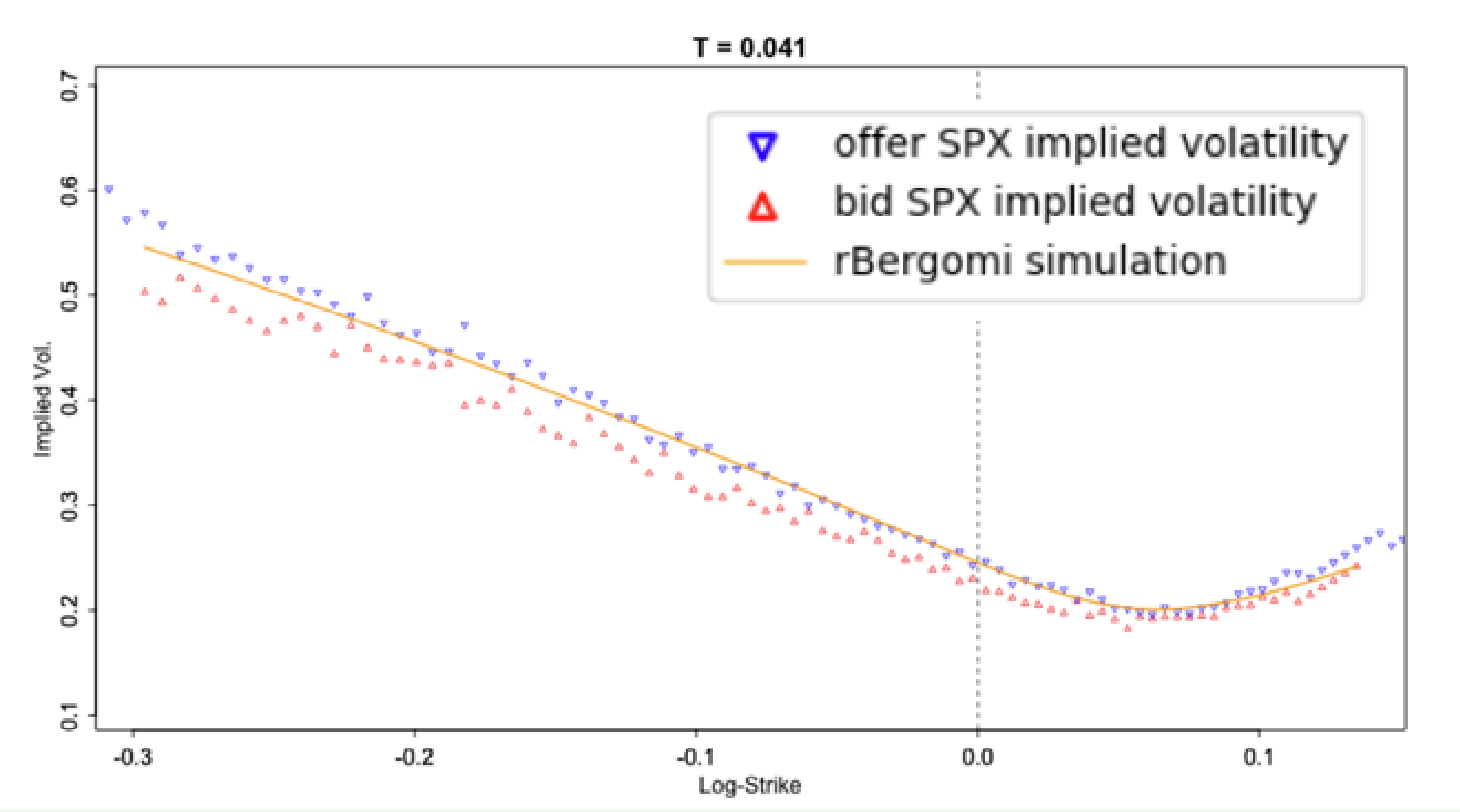


Figure 1: Shortest dated smile as of Feb 4, 2010. The rBergomi model has $H = 0.07$, $\rho = -0.9$, $\eta = 1.9$.

Goals

1. $I(t) \sim \mathcal{N}(0, t^{2H}) \Rightarrow$ exact method available using Cholesky decomposition $\Rightarrow$ requires $\mathcal{O}(n^3)$ offline cost, $\mathcal{O}(n^2)$ computation cost, $\mathcal{O}(n^2)$ storage $\Rightarrow$ resort to approximative method instead.
2. $\xi_0(t)$ can be any given initial forward variance swap curve consistent with market price $\Rightarrow$ can be calibrated using data.

Kernel approximation

- **Discretization**: equidistant grid $0 = t_0 < t_1 < \cdots < t_n = T$ with $\tau = \frac{T}{n}$ and $t_i := i\tau$
- Observe the singularity of $G(t)$ at $t = 0$, one may **keep the exact form of the kernel near singularity** and **apply approximation away from singularity**.

Hybrid scheme

$$\hat{G}_{Hyb}(t) = \begin{cases} t^{H-\frac{1}{2}} & t \in [0, t_\kappa] \\ (b_k^*)^{H-\frac{1}{2}} & t \in [t_{k-1}, t_k],\ k \geq \kappa + 1 \end{cases}$$

$$b_k^* = \tau\left(\frac{k^{H+\frac{1}{2}} - (k-1)^{H+\frac{1}{2}}}{H+\frac{1}{2}}\right)^{1/(H-\frac{1}{2})}$$

Hybrid scheme implements **piecewise constant interpolation** on $[t_\kappa, T]$, b_k^* is optimal point that minimizes the asymptotic MSE induced by the discretization.

Sum of exponentials (SOE) scheme

By Bernstein's theorem, $G(t) = \frac{1}{\Gamma(\frac{1}{2}-H)}\int_0^\infty \mathrm{e}^{-xt}x^{-H-\frac{1}{2}}\mathrm{d}x$

$$\hat{G}_{SOE}(t) = \begin{cases} t^{H-\frac{1}{2}} & t \in [0, \tau) \\ \sum_{j=1}^{N_{\text{exp}}} \omega_j \mathrm{e}^{-\lambda_j t} & t \in [\tau, T] \end{cases}$$

SOE scheme uses **piecewise high order polynomial interpolation** on $[\tau, T]$, where $\lambda_j \geq 0$'s are the interpolation points (nodes) and $\omega_j > 0$'s are the corresponding weights, which can be obtained via **Gaussian quadrature**. It's delicate to find good nodes and weights. There are two possible methods.

1. **Bayer et al.**[2]: applies m-point Gauss-Jacobi quadrature with weight function $x^{-H-\frac{1}{2}}$ to n geometrically spaced intervals $[\xi_i, \xi_{i+1}]_{i=0,\cdots,n-1}$ and use Riemann-type approximation on $[0, \xi_0]$.
2. **Jiang et al.**[3]: applies m_0-point Gauss-Jacobi quadrature with weight function $x^{-H-\frac{1}{2}}$ on $[0, 2^{-M}]$ and m_1-point Gauss-Legendre quadrature to $M+N$ dyadic intervals $[2^i, 2^{i+1}]_{i=-M,\cdots,N-1}$.

Sampling

We use **Jiang et al.'s work** to deal with the fractional kernel, the resulted approximation for $I(t)$ is

$$\hat{I}(t_i) = \sqrt{2H}\sum_{j=1}^{N_{\text{exp}}}\omega_j\int_0^{t_{i-1}}\mathrm{e}^{-\lambda_j(t_i-s)}\mathrm{d}W_s + \sqrt{2H}\int_{t_{i-1}}^{t_i}(t_i-s)^{H-\frac{1}{2}}\mathrm{d}W_s$$

$$=: \sqrt{2H}\sum_{j=1}^{N_{\text{exp}}}\omega_j\hat{I}^j_{\mathcal{F}}(t_i) + I_{\mathcal{N}}(t_i).$$

The **local part** $I_{\mathcal{N}}(t_i) \sim \mathcal{N}(0, \tau^{2H})$ can be simulated exactly; jth component of **history part** $\hat{I}^j_{\mathcal{F}}(t_i)$ has the recursive relation

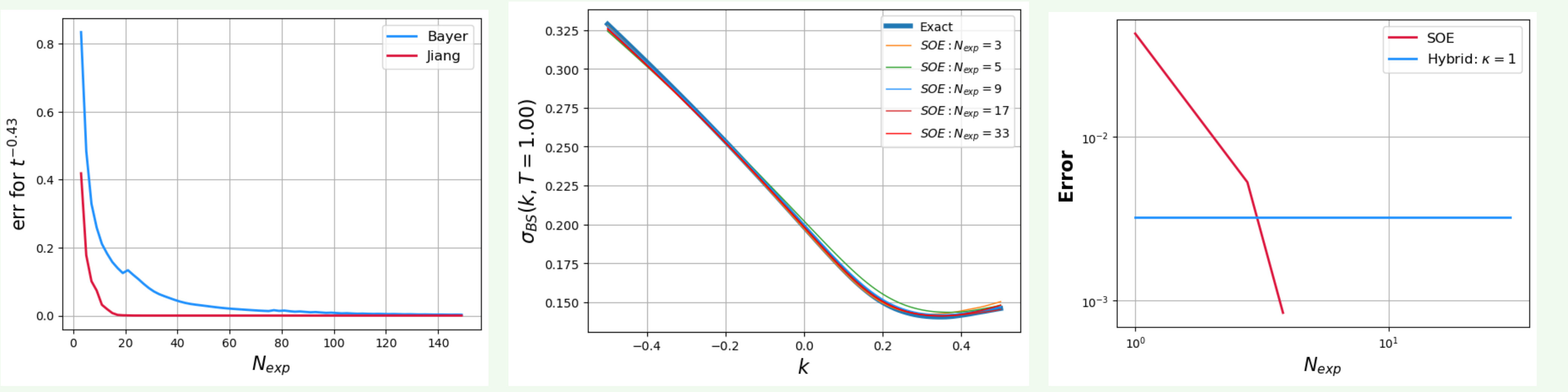
$$\hat{I}^j_{\mathcal{F}}(t_i) = \begin{cases} 0 & i = 1 \\ \mathrm{e}^{-\lambda_j\tau}\left(\hat{I}^j_{\mathcal{F}}(t_{i-1}) + \int_{t_{i-2}}^{t_{i-1}}\mathrm{e}^{-\lambda_j(t_{i-1}-s)}\mathrm{d}W_s\right) & i \geq 2 \end{cases}$$

To simulate S_{t_i} and V_{t_i}, we need the $(N_{\text{exp}}+2)$-dimensional Gaussian vector

$$\left(W_{t_i} - W_{t_{i-1}}, \int_{t_{i-1}}^{t_i}\mathrm{e}^{-\lambda_1(t_i-s)}\mathrm{d}W_s, \cdots, \int_{t_{i-1}}^{t_i}\mathrm{e}^{-\lambda_{N_{\text{exp}}}(t_i-s)}\mathrm{d}W_s, I_{\mathcal{N}}(t_i)\right)$$

whose distribution is independent of i. With samples of V_t available, S_t is solved by Euler-Maruyama method.

For Goal 1, the SOE based simulation scheme only requires $\mathcal{O}(N_{\text{exp}}^3)$ off-line cost, $\mathcal{O}(N_{\text{exp}} n \log n)$ computation cost (achieved by FFT), $\mathcal{O}(nN_{\text{exp}})$ storage. Numerical results indicate $\boldsymbol{N_{\text{exp}} = 3}$ **already produced good results**.



(a) Kernel approximation (b) Implied volatility by SOE (c) Implied volatility error

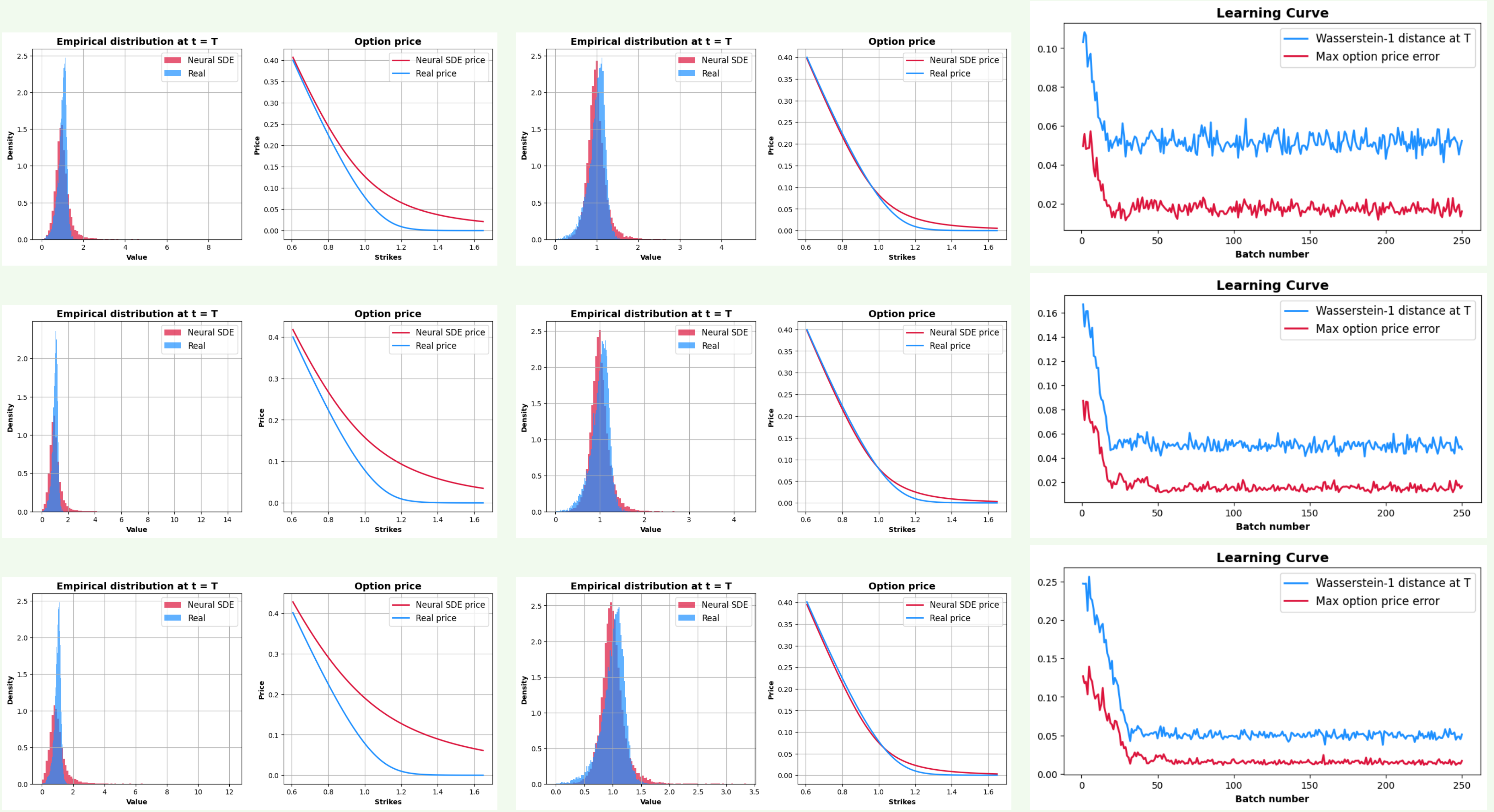
Figure 2: (2(a)) compares the kernel approximation performance between Bayer's scheme and Jiang's, which validates our choice for Jiang. (2(b)) plots the implied volatility of SOE scheme with different N_{exp}s. (2(c)) compares the implied volatility error of SOE scheme and Hybrid scheme.

Neural option pricing

For Goal 2: treat the initial forward variance curve $\xi_0(t)$ as an unknown function which is calibrated using data, i.e. $\xi_0(t) = \xi_0(t;\theta)$, θ are the parameters from a certain neural network, and the training data are generated via the SOE scheme.

- **Wasserstein-1 distance as loss**: Wasserstein-1 distance is applied to the empirical distributions of real samples' and neural SDE's.
- **Upper bound for pricing error**: As the call option payoff $f(x) = (x-K)^+$ is Lipschitz-1 continuous, according to *Kantorovich-Robenstein duality*, Wasserstein-1 distance is a natural upper bound for the pricing error, which is validated by the following figure.

$$|\mathbb{E}f(S(T)) - \mathbb{E}f(\hat{S}(T))| \leq W_1(S(T), \hat{S}(T)) \tag{1}$$



(g) After Initialization (h) After Training (i) Loss-Error

Figure 3: We display the main experiment result of this work. **Each row corresponds to different initial forward variance curves** used for NN training. For each column, (3(g)) shows the empirical distributions at time T and the option prices for several strikes of the neural SDE compared to the real testing samples' before training has begun. This is contrasted with (3(h)) which shows the same plots but after training. (3(i)) shows the Wasserstein-1 distance at time T compared to the maximum error of the option price over each batch during training. These two items are precisely the subjects in (1).

References

[1] Mikkel Bennedsen, Asger Lunde, and Mikko S Pakkanen. "Hybrid scheme for Brownian semistationary processes". In: Finance and Stochastics 21 (2017), pp. 931–965.
[2] Christian Bayer and Simon Breneis. "Markovian approximations of stochastic Volterra equations with the fractional kernel". In: Quantitative Finance 23.1 (2023), pp. 53–70.
[3] Shidong Jiang et al. "Fast evaluation of the Caputo fractional derivative and its applications to fractional diffusion equations". In: Communications in Computational Physics 21.3 (2017), pp. 650–678.