Physics-Informed Koopman Neural Estimation of the Heston Model from High-Frequency Observations

Qiuming Zhu[*1], **Haoran Kou**[*1], **Linyi Qian**[*1], [†], **Chunqi Shi**[*2], **Xianyi Wu**[*1], **Ziwei Zhou**[*3]

[1]School of Statistics, East China Normal University, Shanghai 200062, China
[2]China Pacific Insurance (Group) Co., Ltd., Shanghai 200120, China
[3]School of Statistics and Data Science, Shanghai University of Finance and Economics, Shanghai 200433, China
{52284404008,52284404005,lyqian,xywu}@stu.ecnu.edu.cn, chunqishi@gmail.com, 2024310139@stu.sufe.edu.cn

Abstract

We propose a physics-informed learning framework, called Koopman-PINN, to estimate the parameters of the Heston stochastic volatility model with high-frequency price data in financial markets. The method integrates a nonparametric volatility estimation (known as ART-filter in the literature), moment-based parameter initialization, and a neural Koopman operator constrained by the infinitesimal generator of the underlying stochastic differential equation. By incorporating a generator-based loss, the model bridges Koopman theory and neural modeling to handle partially observed coupled stochastic dynamics in a manner consistent with continuous-time evolution. Across diverse parameter combinations reflecting varying market conditions, Koopman-PINN consistently achieves accurate and robust five-parameter recovery, outperforming existing estimators under a minimal set of initialization assumptions.

Code — https://github.com/zqm0423/Koopman-PINN26

Introduction

Stochastic differential equations (SDEs), as a mathematical tool capable of modeling randomness within dynamic systems, have been extensively used in finance since Black and Scholes (1973), who modeled the dynamics of financial asset prices with geometric Brownian motion, introduced SDEs into the area of financial pricing and established the modeling paradigm of finance analysis in continuous-time. An obvious limit of Black-Scholes' model is its constant volatility, which fails to adequately explain phenomena such as the volatility smile. In order to model more complicated financial derivatives, models for stochastic volatility (SV) are needed as supplements (Taylor 1982). The examples include Scott (1987)'s log-normal model characterizing the logarithm of volatility by an Ornstein–Uhlenbeck process, Stein and Stein (1991) who modeled volatility directly with a Gaussian Ornstein–Uhlenbeck process, Hull and White (1987) who introduced a mean-reverting linear volatility model, among others.

Among all stochastic volatility formulations, the Heston model (Heston 1993) stands out as a prominent example. It has attracted considerable attention from both academic researchers and financial practitioners, owing to its remarkable ability to capture the joint dynamics of asset prices and stochastic volatility. However, the estimation of parameters in the Heston model remains a formidable challenge with only the observable trajectories of asset prices. The obstacles include the latent nature of volatility that complicates likelihood-based inference and the nonlinear coupling between the asset and variance processes which makes decoupling the dynamics quite difficult.

To address the problem of partial observability and nonlinear coupling, Azencott, Ren, and Timofeyev (2020) proposed a nonparametric method (referred to as ART method or ART filter below) that can effectively estimate Heston model's latent volatility using paths of asset prices. It is well known that Koopman's operator theory transfers nonlinear dynamics to linear by means of high dimensional lift space. With Azencott, Ren, and Timofeyev (2020)'s nonparametric method as the front-end, we propose a Koopman-PINN architecture that links the Koopman operator theory with neural networks, so as to decouple the Heston model in a high-dimensional lift space, resulting in accurate estimation of its parameters by learning the temporal dynamics of the system.

To summarize, the main contributions are as follows:

- We propose a Koopman-PINN framework by combining Koopman operator theory with neural networks. Across multiple parameter combinations in the Heston model, our method outperforms existing estimation algorithms and achieves consistently high accuracy.
- Compared with traditional methods that require full initialization of all parameters, our approach only requires initial guesses for a subset (e.g., $\kappa_{\max}$, ρ_0, r_0). This significantly reduces the dependency on prior knowledge and demonstrates strong robustness and practical applicability.
- The modular design of the framework ensures compatibility with diverse volatility inputs and initialization

[*]These authors contributed equally.
[†]Corresponding author.

schemes, thereby allowing for flexible integration and the possibility of adapting it to other partially observed SDE models.

Methodology

Problem Definition

Our starting point is the SDE

$$\mathrm{d}X_t = a\left(X_t;\Theta\right)\mathrm{d}t + b\left(X_t;\Theta\right)\mathrm{d}W_t \tag{1}$$

where $X_t \in \mathbb{R}^d$ denotes the state variable, $a(X_t;\Theta)$ the drift coefficients, $b(X_t;\Theta)$ the diffusion coefficients, W_t a standard Brownian motion, and Θ the model parameters.

As one of the most appealing SDE-based formulations in financial modeling, the Heston stochastic volatility (SV) model, a particular form of the SDE (1), characterizes the joint evolution of the price and stochastic variance of an asset. This is done through a nonlinearly coupled and partially observable system

$$\begin{cases} \mathrm{d}\ln S(t) = \left(r - \dfrac{1}{2}V(t)\right)\mathrm{d}t + \sqrt{V(t)}\mathrm{d}W_1(t) \\ \mathrm{d}V(t) = \kappa(\theta - V(t))\mathrm{d}t + \sigma\sqrt{V(t)}\mathrm{d}W_2(t) \end{cases}, \; t \in [0,T] \tag{2}$$

where $S(t)$ is the asset price, $V(t)$ is the instantaneous variance process, r is the risk-free rate or average return, and W_1 and W_2 are two correlated Brownian motions.

The variance $V(t)$ is further assumed to follow a Cox-Ingersoll-Ross (CIR) process to ensure non-negativity and exhibit mean reversion toward a long-run level θ with a speed κ of mean reversion and a volatility σ of variance ("vol-of-vol").

The Brownian motions $W_1(t)$ and $W_2(t)$ are defined by

$$\begin{cases} W_1(t) = \sqrt{1-\rho^2}W_s(t) + \rho W_v(t), \\ W_2(t) = W_v(t) \end{cases} \tag{3}$$

where $W_s(t)$ and $W_v(t)$ are two independent standard Brownian motions, $\rho \in [-1,1]$ is a constant characterizing the instantaneous correlation between $W_1(t)$ and $W_2(t)$, satisfying

$$dW_1(t)dW_2(t) = \rho dt. \tag{4}$$

This correlation captures the leverage effect widely observed in financial markets.

The objective is to estimate the model parameters $\Theta = (\kappa, \theta, \sigma, \rho, r)$ described above using solely the data of the asset price.

The Koopman-PINN Architecture

It is the latent nature of $V(t)$ and its nonlinear coupling with $S(t)$ that makes estimation of the Heston parameters $\Theta = (\kappa, \theta, \sigma, \rho, r)$ particularly challenging.

We propose Koopman-PINN, a modular framework for estimating Θ from partially observed data. It integrates nonparametric volatility filtering, moment-based initialization, and Koopman-consistent operator learning into a unified pipeline.

As shown in Figure 1, Koopman-PINN comprises two components. The initialization module estimates $\hat{V}k$ via ART filtering and derives initial values for $(\kappa_0, \theta_0, \sigma_0)$ using moment estimators. The main neural module processes inputs $X_k = [\ln S_k, \hat{V}k]$ through a lifting network $F\vartheta(*)$, a learned linear operator A, and a fixed reconstruction matrix C to predict $\hat{X}_{k+1}$. A parallel network $N_\alpha(*)$ learns a residual μ_k, refining the lifted dynamics via $Z_{k+1} = AZ_k + \mu_k$. The model is trained with a composite loss enforcing consistency in both latent evolution and output predictions.

Koopman Theory

Koopman operator theory offers a linear framework for representing the evolution of nonlinear dynamics $\mathbf{x}_{k+1} = \mathbf{F}(\mathbf{x}_k)$ by the evolution of *observables* g rather than the state itself. This is achieved via a linear Koopman operator $\mathcal{K}_{\Delta t}$ acting on the space of functions: $\mathcal{K}_{\Delta t}g(\mathbf{x}_k) = g(\mathbf{F}(\mathbf{x}_k)) = g(\mathbf{x}_{k+1})$, which is linear regardless of the nonlinearity of $\mathbf{F}$.

The operator's spectral properties (eigenvalues λ_j, eigenfunctions ϕ_j, and modes ψ_j) can describe the observable's evolution via the spectral expansion $g(\mathbf{x}_k) = \sum_j \lambda_j^k \psi_j \phi_j(\mathbf{x}_0)$.

In continuous time, letting $\Delta t \to 0$, the family $\{\mathcal{K}_t\}_{t\geq 0}$ generates the infinitesimal operator $\mathcal{K}g := \lim_{t\to 0}(\mathcal{K}_t g - g)/t$. For a flow $\mathbf{x}$ generated by an SDE of the form in Equation (1), the infinitesimal generator $\mathcal{K}$ acts as:

$$\mathcal{K}g(x) = a(x;\Theta)\frac{\mathrm{d}g(x)}{\mathrm{d}x} + \frac{1}{2}b^2(x;\Theta)\frac{\mathrm{d}^2 g(x)}{\mathrm{d}x^2}, \tag{5}$$

capturing drift and diffusion.

Connection to the Algorithm Proposed. In the algorithm described above in Section "The Koopman-PINN Architecture", the state $X_k = \left[\ln S_k, \hat{V}_k\right]$ is lifted to latent observables $Z_k \in \mathbb{R}^m$, the learned linear operator $A \in \mathbb{R}^{m\times m}$ acts as the discrete Koopman operator $\mathcal{K}_{\Delta t}$ whose eigenvalues λ_j encode growth/decay rates, and the fixed reconstruction matrix $C \in \mathbb{R}^{1\times m}$ consists of the Koopman modes ψ_j so that $\hat{X}_{k+1} = CAZ_k$.

Moreover, a parallel residual network $N_\alpha(Z_k)$ is employed to correct the mismatch between the linearly propagated $\hat{Z}_{k+1} = AZ_k$ and the true next latent state Z_{k+1}, and the generator-consistency loss forces A to approximate the SDE 's infinitesimal generator in lifted space. By approximating the lifting by a Network ϕ_θ, the algorithm yielding a data-driven Koopman triple $(\lambda_j, \phi_\theta, C)$ for stochastic-volatility inference.

Algorithm Details

Volatility Filtering and Initialization. We apply the ART scheme (Azencott, Ren, and Timofeyev 2020) to estimate spot volatilities from high-frequency returns using a kernel-weighted realized volatility estimator:

$$\hat{V}_i = \frac{1}{J\Delta_{HF}}\sum_{k=1}^{J} w_k r_{i,k}^2, \qquad w_k = \frac{JK(k)}{\sum_{j=1}^{J}K(j)},$$

where J is the number of high-frequency returns used within each estimation window, Δ_{HF} denotes the high-frequency time step between consecutive log-price observations, and $r_{i,k}$ is the k-th high-frequency log-return within the i-th

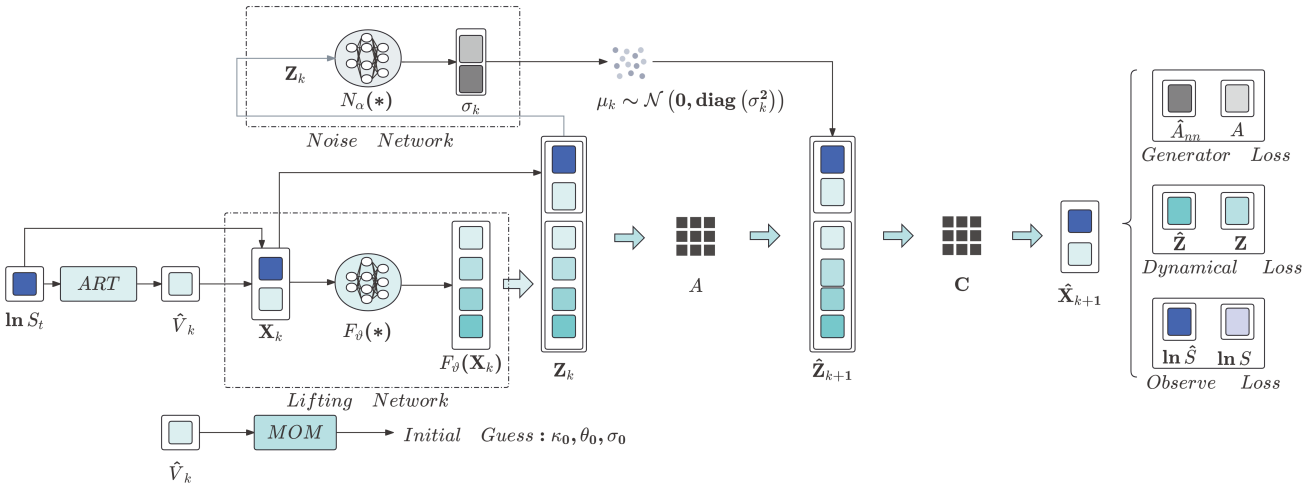

Figure 1: Architecture of the Koopman-PINN Estimation Framework for Latent Stochastic Volatility Models

bucket. The kernel function $K(*)$ assigns weights to returns within each bucket, and w_k are the normalized weights.

Each volatility estimate $\hat{V}_i$ is computed over a window of length $J\Delta_{HF}$. To reduce serial correlation among volatility estimates, the full bucket length is set to $m = J + J_{\text{gap}}$, where J_{gap} defines the number of unused high-frequency returns between buckets. This results in a low-frequency sampling interval of $\Delta_{LF} = m\Delta_{HF}$, and yields a total of $M = \lfloor T/\Delta_{LF} \rfloor$ volatility observations.

Let $B = \lfloor b/\Delta_{LF} \rfloor$ be the lag in low-frequency steps corresponding to a physical lag b, where b denotes a time lag in physical units (e.g., 0.6 days). The CIR parameters are initialized via moment matching as

$$\hat{\kappa} = -\frac{1}{b}\log\left(\frac{\hat{K}(b)}{\hat{K}(0)}\right), \qquad \hat{\theta} = \hat{m}, \qquad \hat{\sigma}^2 = \frac{2\hat{K}(0)\hat{\kappa}}{\hat{\theta}},$$

where $\hat{m} = \frac{1}{M}\sum_{i=1}^{M}\hat{V}_i$ is the sample mean, and

$$\hat{K}(b) = \frac{1}{M-B}\sum_{k=1}^{M-B}\hat{V}_k\hat{V}_{k+B} - \hat{m}^2$$

is the empirical autocovariance of the estimated volatility series at lag b.

This ART-based initialization is non-iterative and provides consistent estimates of the CIR parameters. It also guides the optimal choice of J, Δ_{LF}, and b to ensure low bias and controlled variance in the downstream parameter estimation.

State lifting and projection. Given high-frequency prices $S(t)$, we first compute log prices $\ln S(t)$ and apply the ART filter to estimate the latent volatility path $\hat{V}_k$. At each time step k, we then form a reconstructed state vector:

$$X_k = [\ln S_k, \hat{V}_k],$$

which serves as an input proxy for the true underlying system state. We then use a neural lifting network F_ϑ to embed X_k into a high-dimensional feature space:

$$\xi_k = F_\vartheta(X_k).$$

The lifted observable is formed by concatenating raw states and lifted features:

$$Z_k = [X_k, \xi_k].$$

A fixed projection matrix C recovers the physical state prediction from the lifted space:

$$\hat{X}_{k+1} = C\hat{Z}_{k+1}, \quad \text{with} \quad C = \begin{bmatrix}\mathbf{I}_{2\times 2} & 0 & \cdots & 0\end{bmatrix}.$$

This plays a role reminiscent of Koopman modes by projecting from the function space back to the original state coordinates, but here it is explicitly constructed to select the first two entries.

Learned Koopman operator. We model the latent time evolution using a learned linear operator A with multiplicative stochastic noise:

$$Z_{k+1} = AZ_k + \mu_k, \quad \mu_k = \sigma_k \odot \epsilon_k, \quad \epsilon_k \sim \mathcal{N}(0, I),$$

where the matrix A approximates the Koopman operator acting on the lifted observable Z_k. To account for model uncertainty and stochasticity, we introduce a neural noise network N_α that takes Z_k as input and outputs element-wise noise magnitudes:

$$\sigma_k = N_\alpha(Z_k) \in [0, 1].$$

This boundedness is achieved via a sigmoid activation in the final layer of N_α, which ensures each component of σ_k stays in $[0, 1]$, thus avoiding gradient explosion due to uncontrolled noise amplification.

Loss design. Our total training loss combines three objectives:

Observation loss enforces agreement with the next true log-price:

$$\mathcal{L}_{\text{obs}} = \frac{1}{M}\sum_{k=1}^{M}\left(\ln \hat{S}_{k+1} - \ln S_{k+1}\right)^2.$$

Dynamical consistency loss maintains consistency in the lifted space:

$$\mathcal{L}_{\text{dyn}} = \frac{1}{M}\sum_{k=1}^{M} \|\hat{Z}_{k+1} - Z_{k+1}\|_F^2.$$

PINN-inspired generator loss ensures that the learned operator is consistent with the infinitesimal generator derived from the SDE. As detailed in Section **Koopman Theory**, the generator $\mathcal{K}$ for the Heston model is applied component-wise to the lifted observable Z via Itô's formula.

To enforce generator consistency, we solve a least-squares regression problem of the form:

$$\min_{\mathbf{L}} \|\mathbf{L}Z - \mathcal{K}Z\|_F^2,$$

where $Z \in N \times \mathbb{R}^{d_{Z_k}}$ stacks the lifted observables Z_k across N time steps, and $\mathcal{K}Z$ denotes the application of the Heston infinitesimal generator to each Z_k, computed via automatic differentiation. The matrix $\mathbf{L} \in N \times N$ serves as a finite-dimensional surrogate for the Koopman infinitesimal generator, capturing its linear action on observables within the learned subspace. That is, for any lifted observable Z_k, we require $\mathcal{K}Z_k \approx \mathbf{L}Z_k$. The corresponding infinitesimal generator form can be derived via Itô's lemma, yielding:

$$\begin{aligned}\mathcal{K}f(\ln S, V) = \Big(r - \frac{V}{2}\Big)\frac{\partial f}{\partial \ln S} + \kappa(\theta - V)\frac{\partial f}{\partial V} \\ + \frac{1}{2}V\frac{\partial^2 f}{\partial \ln S^2} + \frac{1}{2}\sigma^2 V\frac{\partial^2 f}{\partial V^2} + \rho\sigma V\frac{\partial^2 f}{\partial \ln S \partial V}. \quad (6)\end{aligned}$$

Here, f represents a Koopman observable, instantiated in our implementation as each lifted coordinate Z_k at time step k. This formulation aligns with the infinitesimal generator framework presented in Section **Koopman Theory**, ensuring that the learned generator matrix $\mathbf{L}$ captures the local differential structure imposed by the SDE dynamics.

While the learned generator matrix $\mathbf{L}$ captures the infinitesimal evolution of observables in the lifted space, practical inference and prediction require modeling their progression over finite time intervals. To bridge this gap, we construct a generator-consistent transition operator via the matrix exponential:

$$A_{\text{gen}} = \exp(\Delta t \mathbf{L}).$$

This construction follows from the fundamental relationship between infinitesimal generators and semigroups of operators in continuous-time dynamics, where $\exp(\Delta t\mathbf{L})$ defines the Koopman operator over a finite horizon Δt (Mauroy, Susuki, and Mezic 2020). In our framework, A_{gen} thus provides a principled approximation to the true Koopman operator acting on observables over discrete time steps.

Since the Koopman-PINN simultaneously learns a transition matrix A_{nn} from data, we enforce consistency by penalizing the discrepancy between this learned operator and the generator-induced one:

$$\mathcal{L}_{\text{gen}} = \frac{1}{M^2}\|A_{\text{nn}} - A_{\text{gen}}\|_F^2.$$

Algorithm 1: Koopman-PINN Parameter Estimation

Input: High-frequency log-prices $\ln S(t)$
Preprocessing: Estimate $\hat{V}_t$ via ART filter on low-frequency buckets.
Initialization: Use moment-of-moments (MoM) to compute initial guesses $(\kappa_0, \theta_0, \sigma_0)$
Fixed Inputs: $(\kappa_{\max}, r_0, \rho_0)$ manually specified
Output: Estimated parameters $(\hat{\kappa}, \hat{\theta}, \hat{\sigma}, \hat{\rho}, \hat{r})$

1: Construct state $X_k = [\ln S_k, \hat{V}_k]$
2: Compute lifted observables $Z_k = [X_k, F_\vartheta(X_k)]$ using the lifting network
3: Compute additive noise term μ_k via noise network
4: Predict next lifted state $\hat{Z}_{k+1} = AZ_k + \mu_k$
5: Compute generator derivative $\mathcal{K}Z_k$ using Itô formula and automatic differentiation
6: Estimate generator matrix $\mathbf{L}$ by solving $\min_{\mathbf{L}} \|\mathbf{L}Z - \mathcal{K}Z\|_F^2$
7: Compute generator-consistent linear operator $A_{\text{gen}} = \exp(\Delta t\, \mathbf{L})$
8: Compute losses:
- $\mathcal{L}_{\text{obs}} = \frac{1}{M}\sum_{k=1}^{M}\big(\ln \hat{S}_{k+1} - \ln S_{k+1}\big)^2$.
- $\mathcal{L}_{\text{dyn}} = \frac{1}{M}\sum_{k=1}^{M}\|\hat{Z}_{k+1} - Z_{k+1}\|_F^2$
- $\mathcal{L}_{\text{gen}} = \frac{1}{M^2}\|A_{\text{nn}} - A_{\text{gen}}\|_F^2$

9: $\mathcal{L}_{\text{total}} = \lambda_{\text{obs}}\mathcal{L}_{\text{obs}} + \lambda_{\text{dyn}}\mathcal{L}_{\text{dyn}} + \lambda_{\text{gen}}\mathcal{L}_{\text{gen}}$ with GradNorm reweighting
10: Update network and parameter estimates: $(\hat{\kappa}, \hat{\theta}, \hat{\sigma}, \hat{\rho}, \hat{r})$

This alignment ensures that both learned and generator-induced dynamics match in discrete time.The total loss is then

$$\mathcal{L}_{\text{total}} = \lambda_{\text{obs}}\mathcal{L}_{\text{obs}} + \lambda_{\text{dyn}}\mathcal{L}_{\text{dyn}} + \lambda_{\text{gen}}\mathcal{L}_{\text{gen}},$$

with hyperparameters $\lambda_{\text{obs}}, \lambda_{\text{dyn}}, \lambda_{\text{gen}}$ balancing each term.

To summarize, our algorithm realizes a principled integration of Koopman operator theory and physics-informed deep learning. The generator loss $\mathcal{L}_{\text{gen}}$ plays a central role by enforcing infinitesimal consistency between the learned dynamics and the underlying Heston SDE, beyond mere trajectory fitting. By constructing a generator-consistent operator A_{gen} from the estimated infinitesimal generator matrix $\mathbf{L}$—obtained via least-squares alignment between the generator action $\mathcal{K}Z$ and the lifted observables Z—and aligning it with the network-predicted A_{nn}, the model ensures that both local differential behavior and global time-evolution are faithfully captured. This connection grounds the learning procedure in the rigorous semantics of stochastic differential equations, and allows for stable and interpretable parameter estimation under uncertainty.

Experiments

We evaluate Koopman-PINN on simulated Heston trajectories across diverse parameter regimes, covering both partial and full inference tasks. Comparisons with classical estimators demonstrate its robustness and accuracy. We also assess different volatility filters and find ART to be the most effective. All experiments use 500 trajectories per configura-

Case	κ	θ	σ	ρ	r
A	1.0	0.04	0.20	0.0	0.08
B	3.0	0.04	0.20	-0.5	0.08
C	2.0	0.10	0.30	0.0	0.05
D	2.0	0.08	0.50	0.8	0.08
E	1.2	0.04	0.30	-0.8	0.02
F	2.0	0.04	0.30	0.0	0.08

Table 1: Six parameter configurations used in experiments

tion (six total), and results are reported as average MSE and MAE.

Datasets and Settings

To evaluate the Koopman-PINN framework under controlled conditions, we generate synthetic datasets by discretizing the Heston stochastic volatility model using the Euler–Maruyama scheme. Each simulated trajectory consists of a two-dimensional time series describing the evolution of the log-price $\ln S_t$ and the latent variance V_t. The discretization equations take the form:

$$\begin{cases} \ln S_{k+1} = \ln S_k + \left(r - \frac{1}{2}V_k\right)\Delta t + \sqrt{V_k}\,\Delta W_{1_k}, \\ V_{k+1} = V_k + \kappa(\theta - V_k)\Delta t + \sigma\sqrt{V_k}\,\Delta W_{2_k}, \end{cases} \tag{7}$$

where $\Delta W_{1_k} = \sqrt{1-\rho^2}\Delta W_{s_k} + \rho\Delta W_{v_k}$ and $\Delta W_{2_k} = \Delta W_{v_k}$, $\Delta W_{s_k}, \Delta W_{v_k} \sim \mathcal{N}(0, \Delta_{HF})$ are independent Brownian increments, and the correlation between the asset and variance processes is introduced via the parameter ρ. We discretize the interval $[0, T]$ into N uniform steps of size $\Delta t = T/N$, where T denotes the total time horizon and N the number of time steps per trajectory.

Our experiments are conducted under six parameter configurations summarized in Table 1. For all these settings, trajectories are initialized with the same asset price and volatility values, namely $S_0 = 4.605$ and $V_0 = 0.04$.

We set the total time horizon to $T = 3$, and simulate 500 independent trajectories, each consisting of $N = 15,000$ discrete time steps. These settings ensure both a sufficiently high-frequency resolution for volatility estimation and a diverse collection of sample paths for robust performance assessment.

For all experiments, the latent variance series is preprocessed using the ART filter (Azencott, Ren, and Timofeyev 2020), unless otherwise noted. The high-frequency log-returns are partitioned into non-overlapping buckets of width $\Delta_{LF} = 0.04$, each containing $m = 20$ returns sampled at interval $\Delta_{HF} = 0.002$. Within each bucket, we use $J = J_{\text{opt}} = 10$ returns. The ART filter produces low-frequency estimates of V_t, which serve as inputs to all subsequent estimation modules.

Estimating Volatility Parameters (κ, θ, σ) with Fixed (r, ρ)

We begin by evaluating the performance of Koopman-PINN under a simplified setting in which only the variance-related parameters (κ, θ, σ) are estimated, while the correlation ρ and risk-free rate r are fixed at their ground-truth values. This design isolates the latent variance process V_t—assumed to follow a CIR model—and allows for focused comparison across parameter estimation methods.

All estimators follow an identical three-stage pipeline:

1. estimate the latent volatility path $\hat{V}_t$ from observed log-prices using the ART filter (Azencott, Ren, and Timofeyev 2020);
2. initialize (κ, θ, σ) using robust moment-based estimates computed from three trajectory segments;
3. estimate final parameters using either Koopman-PINN, GMM, QMLE, or EMLE.

To mitigate the known upward bias of ART filtering on κ (Okhrin, Rockinger, and Schmid 2025), we compute the median estimate κ_{med} over three equal-length segments of each trajectory. A conservative upper bound $\kappa_{\max}$ is then applied, and the initial value for κ is chosen as the smaller of κ_{med} and $\kappa_{\max}$:

$$\kappa_0 = \min\left(\kappa_{\text{med}}, \kappa_{\max}\right)$$

This upper bound acts as a regularization device to suppress spurious overestimation during initialization. Among all methods evaluated, only Koopman-PINN leverages this single capped initialization strategy; in contrast, classical estimators such as GMM, QMLE, and EMLE require full initial guesses for all parameters (κ, θ, σ) and are typically more sensitive to initialization quality.

Table 2 compares estimation accuracy for (κ, θ, σ) (with r, ρ fixed), using ART-filtered volatility and MoM initialization. Our Koopman-PINN (KP) method consistently outperforms classical methods like GMM, QMLE and EMLE. These classical approaches, which discard up to 5% of outliers (e.g., diverging $\kappa > 10$) yet still suffer high errors—especially in estimating κ. In contrast, Koopman-PINN yields robust, accurate estimates without trimming, demonstrating superior generalization.

Full Parameter Estimation $(\kappa, \theta, \sigma, \rho, r)$

We extend estimation to the full parameter set $(\kappa, \theta, \sigma, \rho, r)$, maintaining the ART-MoM pipeline. Variance parameters are initialized via capped median-moment estimates, and (r_0, ρ_0) are manually set across six configurations reflecting different prior assumptions.

Table 3 and 4 compares the performance of Koopman-PINN against three classical estimators—EKF-ML, Joint-Lik(J-Lik), and Composite-Lik(C-Lik) under these six parameter configurations (Cases A–F). Koopman-PINN consistently achieves the lowest total MSE and MAE across nearly all cases, especially in estimating σ, where Koopman-PINN outperforms all baseline methods and achieves the most accurate estimates across all six parameter configurations.

While classical methods (e.g., QMLE, EMLE) are sensitive to misspecification and yield unstable estimates—especially for large κ—Koopman-PINN remains robust, even with coarse initializations of r and ρ. This reflects

		Case A			Case B			Case C			Case D			Case E			Case F		
Meth.	**Met.** ↓	κ	θ	σ	κ	θ	σ	κ	θ	σ	κ	θ	σ	κ	θ	σ	κ	θ	σ
KP	MSE	**0.234**	**0.001**	**0.002**	**0.851**	**0.001**	**0.008**	**0.218**	**0.001**	**0.007**	**0.218**	0.003	**0.015**	**0.229**	0.002	**0.007**	**0.226**	0.001	**0.004**
	MAE	**0.483**	**0.026**	**0.033**	**0.882**	**0.025**	**0.081**	**0.458**	**0.030**	**0.070**	**0.461**	**0.039**	**0.105**	**0.477**	0.034	**0.074**	**0.465**	0.029	**0.051**
GMM	MSE	33.4	**0.001**	0.029	34.2	**0.000**	0.030	27.0	0.003	0.054	22.2	**0.002**	0.103	26.7	**0.001**	0.043	28.3	**0.001**	0.046
	MAE	4.472	**0.022**	0.144	5.419	**0.019**	0.157	4.095	0.055	0.202	3.32	0.047	0.256	3.63	**0.024**	0.170	4.13	**0.022**	0.183
QML	MSE	77.9	0.047	2.78	48.9	0.028	3.01	53.6	0.302	2.89	16.4	0.609	2.23	40.2	0.148	2.38	55.1	0.096	2.58
	MAE	8.72	0.155	1.656	6.99	0.159	1.731	7.06	0.497	1.700	2.86	0.696	1.493	5.06	0.216	1.517	7.07	0.184	1.598
EML	MSE	80.4	0.021	2.49	48.9	0.026	2.74	64.0	0.026	2.89	63.8	0.016	2.17	74.4	0.017	2.06	63.6	0.019	2.22
	MAE	8.96	0.136	1.557	6.99	0.158	1.648	8.00	0.160	1.700	7.98	0.117	1.471	8.52	0.106	1.404	7.97	0.127	1.471

Table 2: Estimation performance under six distinct parameter sets. The *Params* column lists (κ, θ, σ), and each cell in Cases A–F reports the corresponding numeric errors in the same order. Lower MSE/MAE indicate better accuracy.

		Case A					Case B					Case C				
Meth.	**Met.** ↓	κ	θ	σ	ρ	r	κ	θ	σ	ρ	r	κ	θ	σ	ρ	r
KP	MSE	0.233	**0.001**	**0.002**	0.159	**0.002**	**0.852**	**0.001**	**0.008**	**0.160**	**0.002**	**0.217**	**0.002**	**0.007**	0.158	**0.002**
	MAE	0.482	**0.024**	**0.032**	0.398	**0.032**	**0.882**	**0.025**	**0.080**	0.400	**0.034**	**0.457**	**0.031**	**0.070**	0.397	**0.038**
EKF	MSE	**0.223**	47.4	0.583	0.226	0.037	1.698	0.032	0.760	0.162	0.015	0.624	20.8	0.460	0.292	0.065
	MAE	**0.430**	0.635	0.345	0.366	0.128	1.127	0.026	0.734	**0.342**	0.092	0.650	0.681	0.447	0.432	0.141
J–Lik	MSE	81.0	0.027	1.099	0.001	0.007	48.9	0.029	1.127	0.260	0.011	64.0	0.120	1.694	0.000	0.014
	MAE	9.00	0.136	1.036	0.017	0.068	6.99	0.165	1.059	0.510	0.084	8.00	0.328	1.296	0.012	0.087
C–Lik	MSE	1.001	0.026	3.240	**0.000**	0.005	9.000	0.026	3.240	0.250	0.004	4.000	0.010	2.890	**0.000**	**0.002**
	MAE	1.000	0.160	1.800	**0.001**	0.072	3.000	0.160	1.800	0.500	0.055	2.000	0.100	1.700	**0.000**	0.044

Table 3: Estimation errors (MSE/MAE) for Cases A–C of the full Heston parameter vector $(\kappa, \theta, \sigma, \rho, r)$. Bold = best.

		Case D					Case E					Case F				
Meth.	**Met.** ↓	κ	θ	σ	ρ	r	κ	θ	σ	ρ	r	κ	θ	σ	ρ	r
KP	MSE	**0.218**	**0.003**	**0.016**	**0.163**	**0.002**	**0.229**	**0.002**	**0.007**	**0.160**	0.005	**0.226**	**0.001**	**0.004**	0.158	**0.002**
	MAE	**0.461**	**0.040**	**0.106**	**0.403**	**0.033**	**0.477**	**0.034**	**0.074**	**0.400**	0.064	**0.465**	**0.028**	**0.050**	0.397	**0.032**
EKF	MSE	0.525	62.1	0.763	0.232	0.158	0.290	0.924	2.437	0.297	0.020	0.707	17.6	0.361	0.239	0.056
	MAE	0.612	0.999	0.550	**0.357**	0.212	0.478	0.117	0.635	0.449	0.059	0.709	0.404	0.371	0.369	0.139
J–Lik	MSE	62.6	0.036	1.104	0.564	0.013	68.8	0.076	1.466	0.647	**0.005**	63.6	0.018	0.937	**0.000**	0.006
	MAE	7.89	0.127	1.033	0.751	0.084	8.18	0.190	1.192	0.804	**0.048**	7.97	0.108	0.958	**0.000**	0.065
C–Lik	MSE	4.00	0.014	2.250	0.640	0.006	1.44	0.026	2.890	0.639	0.021	1.20	0.160	1.700	0.800	0.136
	MAE	2.00	0.120	1.500	0.800	0.076	1.20	0.160	1.700	0.800	0.136	2.00	0.160	1.700	**0.000**	0.072

Table 4: Estimation errors (MSE/MAE) for Cases D–F of the Heston parameter vector $(\kappa, \theta, \sigma, \rho, r)$. Bold = best.

real-world conditions, where r is fixed from market data and ρ is roughly calibrated. By leveraging physics-informed structure, our method refines such priors with minimal manual tuning. These results confirm Koopman-PINN's reliability and stability across regimes, consistently surpassing classical estimators in both accuracy and robustness.

Impact of Volatility Filtering Methods on Estimation Accuracy

While Koopman-PINN only requires coarse volatility inputs, the choice of pre-filtering can significantly affect parameter estimation. We assess this impact in **Case F** by comparing several common filtering methods. Results in Table 5 highlight their influence on estimation accuracy.

- **Rolling Window Realized Volatility (RW-RV)**: Constructed using a sliding window of 20 high-frequency returns ($\Delta_{\text{HF}} = 0.002$), matching the bucket scale of ART filtering.
- **Exponentially Weighted Moving Average (EWMA)**: We evaluate two decay factors—$\lambda = 0.94$ (the standard RiskMetrics value) and $\lambda = 0.97$ (a smoother variant)—to investigate the noise–lag trade-off.
- **GARCH(1,1)**: Fitted on low-frequency returns ($\Delta_{\text{LF}} = 0.04$) via MLE and transformed into daily conditional variances.

Throughout all trials, we keep the moment-based initialization and Koopman-PINN architecture fixed to ensure fair comparison.

Filter	Metric ↓	κ	θ	σ	ρ	r
ART	MSE	**0.226**	**0.001**	**0.004**	**0.158**	**0.002**
	MAE	**0.465**	**0.028**	**0.050**	**0.397**	**0.032**
RW-RV	MSE	0.259	0.001	0.008	0.158	0.002
	MAE	0.481	0.028	0.074	0.398	0.035
EWMA-0.94	MSE	0.291	0.002	0.011	0.160	0.002
	MAE	0.494	0.031	0.091	0.400	0.034
EWMA-0.97	MSE	0.336	0.001	0.015	0.159	0.002
	MAE	0.515	0.028	0.110	0.399	0.033
GARCH(1,1)	MSE	0.434	0.001	0.025	0.158	0.002
	MAE	0.556	0.029	0.142	0.398	0.036

Table 5: Parameter estimation errors (MSE and MAE) under different volatility filtering methods on **Case F**. Lower values(↓) indicate better accuracy.

Table 5 shows that ART filtering achieves the most accurate and stable parameter estimates, with the lowest MSE and MAE across all five parameters. In contrast, EWMA and GARCH inputs introduce larger biases—especially for σ and κ—due to sensitivity to smoothing and frequency mismatches. These results highlight the empirical advantage of ART in extracting informative volatility features under noise and indirect observability.

Related Work

State Inference

In Heston-type models, latent volatility is typically recovered using two classes of methods.

The first includes recursive state-space methods such as the Extended Kalman Filter (EKF) (Javaheri, Lautier, and Galli 2003), Unscented Kalman Filter (UKF) (Wan and Van Der Merwe 2000), and Consistent EKF (Wang et al. 2018), which iteratively estimate latent trajectories via predict-update schemes under known dynamics. The second includes empirical filters widely used in finance, such as Rolling Window Realized Volatility (Blair, Poon, and Taylor 2001), Exponentially Weighted Moving Average (Lucas and Saccucci 1990), and Generalized Autoregressive Conditional Heteroskedasticity (Bollerslev 1986). These directly estimate volatility from returns without modeling latent dynamics, but are sensitive to noise and sampling frequency (Gatheral 2011).

Nonparametric methods like ART (Azencott, Ren, and Timofeyev 2020) can also be broadly viewed as volatility filters. We adopt ART in our pipeline to extract stable inputs before parameter estimation.

Parameter Estimation

Traditional parameter estimation for SDEs typically falls into two categories: "likelihood-based" methods and "sample DNA matching" methods (Hurn, Jeisman, and Lindsay 2007).

The classical likelihood-based approach is exact maximum likelihood estimation (EMLE) (Aït-Sahalia 2002). Because the Heston model is only partially observable and its joint transition density lacks a closed-form Fokker–Planck solution, EMLE is infeasible. Pseudo-maximum likelihood estimation (PMLE) (Tang and Chen 2009) was therefore introduced, substituting an approximating density for the true likelihood, but it cannot identify the correlation ρ between asset returns and volatility innovations. Quasi-maximum likelihood estimation (QMLE) (Nelson 1988) is easy to implement yet prone to local optima in high-dimensional or non-convex likelihood surfaces, limiting its practical reliability.

The sample DNA method's main idea is feature matching. Representative methods include the Generalized Method of Moments(GMM) (Hansen 1982) and Monto Carlo Markov chain(MCMC) (Jacquier, Polson, and Rossi 2004), but both entail high computational costs.

Dynamics Learning

From a dynamical systems perspective, methods that capture system evolution have shown strong potential for parameter estimation in PDEs and SDEs. A representative example is Yan et al. (2025), who combine Koopman theory with neural networks to infer latent states and estimate parameters in partially observed chemical reaction systems governed by PDEs.

Another prominent direction is Physics-Informed Neural Networks (PINNs)(Raissi, Perdikaris, and Karniadakis 2019; Karniadakis et al. 2021), which embed physics information into the loss function, allowing for both forward and inverse problem solving.

Recent research favors modular frameworks that combine state inference, physical constraints, and dynamic prediction to improve generalization and interpretability (Willard et al. 2020). Koopman-PINN follows this principle, unifying these components into a flexible pipeline for SDE parameter estimation.

Conclusion

Koopman-PINN estimates Heston model parameters from high-frequency data by integrating ART filtering, moment-based initialization, and a neural operator consistent with the infinitesimal generator of the underlying SDE. This principled fusion of statistical inference and operator-theoretic learning enables accurate, robust estimation. Empirical results show that Koopman-PINN consistently outperforms classical estimators (GMM, QMLE, EMLE), even under coarse or misspecified initializations. The generator loss enforces local consistency with continuous-time dynamics while supporting flexible learning from discrete data, and the learned operator approximates the Koopman semigroup, enabling generalization across time scales.

Acknowledgments

This work has been partially supported by the National Natural Science Foundation of China (12171158, 12571510) and the Fundamental Research Funds for the Central Universities(2025ECNU-WLJC006).

References

Aït-Sahalia, Y. 2002. Maximum likelihood estimation of discretely sampled diffusions: a closed-form approximation approach. *Econometrica*, 70(1): 223–262.

Azencott, R.; Ren, P.; and Timofeyev, I. 2020. Realised volatility and parametric estimation of Heston SDEs. *Finance and Stochastics*, 24(3): 723–755.

Black, F.; and Scholes, M. 1973. The pricing of options and corporate liabilities. *Journal of political economy*, 81(3): 637–654.

Blair, B. J.; Poon, S.-H.; and Taylor, S. J. 2001. Forecasting S&P 100 volatility: the incremental information content of implied volatilities and high-frequency index returns. *Journal of Econometrics*, 105(1): 5–26. Forecasting and empirical methods in finance and macroeconomics.

Bollerslev, T. 1986. Generalized autoregressive conditional heteroskedasticity. *Journal of econometrics*, 31(3): 307–327.

Gatheral, J. 2011. *The volatility surface: a practitioner's guide*. John Wiley & Sons.

Hansen, L. P. 1982. Large sample properties of generalized method of moments estimators. *Econometrica: Journal of the econometric society*, 1029–1054.

Heston, S. L. 1993. A closed-form solution for options with stochastic volatility with applications to bond and currency options. *The review of financial studies*, 6(2): 327–343.

Hull, J.; and White, A. 1987. The pricing of options on assets with stochastic volatilities. *The journal of finance*, 42(2): 281–300.

Hurn, A. S.; Jeisman, J.; and Lindsay, K. A. 2007. Seeing the wood for the trees: A critical evaluation of methods to estimate the parameters of stochastic differential equations. *Journal of Financial Econometrics*, 5(3): 390–455.

Jacquier, E.; Polson, N. G.; and Rossi, P. E. 2004. Bayesian analysis of stochastic volatility models with fat-tails and correlated errors. *Journal of Econometrics*, 122(1): 185–212.

Javaheri, A.; Lautier, D.; and Galli, A. 2003. Filtering in finance. *Wilmott*, 3: 67–83.

Karniadakis, G. E.; Kevrekidis, I. G.; Lu, L.; Perdikaris, P.; Wang, S.; and Yang, L. 2021. Physics-informed machine learning. *Nature Reviews Physics*, 3(6): 422–440.

Lucas, J. M.; and Saccucci, M. S. 1990. Exponentially weighted moving average control schemes: properties and enhancements. *Technometrics*, 32(1): 1–12.

Mauroy, A.; Susuki, Y.; and Mezic, I. 2020. *Koopman operator in systems and control*, volume 7. Springer.

Nelson, D. B. 1988. *The time series behavior of stock market volatility and returns*. Ph.D. thesis, Massachusetts Institute of Technology.

Okhrin, O.; Rockinger, M.; and Schmid, M. 2025. Observations concerning the estimation of Heston's stochastic volatility model using HF data. *Statistical Papers*, 66(4): 1–23.

Raissi, M.; Perdikaris, P.; and Karniadakis, G. E. 2019. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational physics*, 378: 686–707.

Scott, L. O. 1987. Option pricing when the variance changes randomly: Theory, estimation, and an application. *Journal of Financial and Quantitative analysis*, 22(4): 419–438.

Stein, E. M.; and Stein, J. C. 1991. Stock price distributions with stochastic volatility: an analytic approach. *The review of financial studies*, 4(4): 727–752.

Tang, C. Y.; and Chen, S. X. 2009. Parameter estimation and bias correction for diffusion processes. *Journal of Econometrics*, 149(1): 65–81.

Taylor, S. J. 1982. Financial returns modelled by the product of two stochastic processes-a study of the daily sugar prices 1961-75. *Time series analysis: theory and practice*, 1: 203–226.

Wan, E. A.; and Van Der Merwe, R. 2000. The unscented Kalman filter for nonlinear estimation. In *Proceedings of the IEEE 2000 adaptive systems for signal processing, communications, and control symposium (Cat. No. 00EX373)*, 153–158. Ieee.

Wang, X.; He, X.; Bao, Y.; and Zhao, Y. 2018. Parameter estimates of Heston stochastic volatility model with MLE and consistent EKF algorithm. *Science China Information Sciences*, 61: 1–17.

Willard, J.; Jia, X.; Xu, S.; Steinbach, M.; and Kumar, V. 2020. Integrating physics-based modeling with machine learning: A survey. *arXiv preprint arXiv:2003.04919*, 1(1): 1–34.

Yan, M.; Han, M.; Law, A. W.-K.; and Yin, X. 2025. Self-tuning moving horizon estimation of nonlinear systems via physics-informed machine learning Koopman modeling. *AIChE Journal*, 71(2): e18649.