AIMS Mathematics, 11(11): 38933–38962.
DOI: 10.3934/math.20261501
Received: 11 September 2026
Revised: 10 November 2026
Accepted: 18 November 2026
Published: 23 November 2026

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

Research article

FiLM-MM-PINN: A sensitivity-regularized physics-informed neural network for Heston option pricing and market calibration

Jina Li*, Daiyuan Jia, Yanpei Zhen, and Yaxin Li

Zhongyuan University of Technology, Zhengzhou, Henan 450007, China

* **Correspondence:** Email: lijina@zut.edu.cn.

Abstract: Repeatedly evaluating option prices is required both for option valuation and for market calibration under the Heston model. To provide a reusable pricing model, we proposed a sensitivity-regularized physics-informed neural network, termed FiLM-MM-PINN, that integrates a moment-matching prior with feature-wise linear modulation. In the forward problem, a forward-normalized reformulation of the Heston partial differential equation was imposed as a physical constraint, yielding a unified pricing surrogate applicable within a prescribed state–parameter domain, with terminal and boundary conditions enforced and without reference option prices as training labels. The network employs the moment-matched Black–Scholes price as a prior and combines parameter encoding, operator-related features, and feature-wise linear modulation to learn price corrections; local parameter sensitivity constraints were further introduced to regularize parameter derivatives. In the inverse problem, the trained network weights were frozen and the Heston parameters, including the initial variance, were estimated from S&P 500 index option quotes via multi-start, multi-stage optimization. Numerical experiments evaluated pricing accuracy, parameter sensitivities, and synthetic parameter recovery. Ablation studies supported the respective contributions of the moment-matching prior, the operator-related features, and the feature-wise linear modulation to pricing accuracy and to the reduction of equation residuals. On an independent market validation set, the relative L^2 error was 0.3278% and the coefficient of determination was 0.999968; the market fit error was lower than those obtained with the Fourier-cosine expansion method and the alternating direction implicit finite-difference method used as calibration benchmarks under the adopted calibration protocol. These results support the use of the proposed method for physics-constrained parametric pricing and market calibration through a frozen, differentiable pricing model.

Keywords: Heston model; physics-informed neural network; moment matching; feature-wise linear modulation; market calibration
Mathematics Subject Classification: 91G20, 65M06, 68T07

1. Introduction

Option pricing under stochastic volatility is a representative parameter-dependent partial differential equation (PDE) problem in computational finance. The Black–Scholes model and Merton's continuous-time framework established the classical theory of option valuation under no arbitrage, continuous trading, and constant volatility [1, 2]. However, the constant-volatility assumption cannot reproduce the term structure and asymmetric shapes of implied volatility observed in equity option markets. The Heston model addressed this limitation by introducing a mean-reverting stochastic variance process correlated with the asset return [3, 4]. Although European option prices under the Heston model can be expressed through its characteristic function, practical valuation still requires numerical integration or approximation. The fast Fourier transform (FFT) and Fourier–cosine expansion (COS) methods provide efficient approaches when the characteristic function is available [5, 6], while careful treatment is required to avoid numerical instabilities associated with complex-valued characteristic functions [7]. Finite-difference methods solve the corresponding pricing equation directly, and alternating direction implicit (ADI) schemes are particularly suitable for handling its mixed-derivative term [8–10]. Monte Carlo methods offer greater flexibility for pricing complex derivatives [11], but their efficiency depends strongly on the selected discretization and simulation schemes [12, 13].

These conventional methods are effective for a fixed set of Heston parameters. Market calibration, however, requires option prices to be evaluated repeatedly as the mean-reversion rate, long-run variance, volatility of variance, correlation coefficient, and initial variance are updated. The repeated solution of the pricing problem may therefore become computationally expensive, particularly when multiple initial points, maturities, and strikes are considered. Efficient calibration algorithms based on characteristic functions and analytical gradients can reduce this cost [14], but their implementation remains closely tied to the selected numerical pricing formula. More generally, the Heston pricing problem is governed by a degenerate parabolic equation whose coefficients depend jointly on the parameter vector $\boldsymbol{\mu} = (\kappa, \theta, \xi, \rho)$. Calibration consequently requires a numerical model that approximates the parameter-to-solution relationship over a prescribed parameter domain rather than a single solution associated with fixed parameters. Moreover, a model used in gradient-based calibration must capture not only option values but also their local variations with respect to the Heston parameters.

Deep-learning-based PDE solvers provide a mesh-free alternative for approximating parameter-dependent solutions. The deep Galerkin method and the deep backward stochastic differential equation method construct neural approximations using equation residuals and stochastic representations, respectively [15, 16]. Physics-informed neural networks incorporate governing equations and initial or boundary conditions into the training objective, allowing forward and inverse differential-equation problems to be treated within a common computational framework [17, 18]. Subsequent developments include domain-decomposition architectures, general-purpose computational libraries, and systematic studies of physics-informed scientific machine learning [19–21]. Nevertheless, the simultaneous optimization of multiple physical constraints can produce imbalanced gradients and difficult loss landscapes, leading to inaccurate or unstable convergence [22–24]. Adaptive weighting [25] and sampling strategies can alleviate some of these difficulties [26–28], but learning a family of solutions over both the state and parameter domains

remains challenging.

Physics-informed neural networks have increasingly been applied to European and American option pricing [29, 30], local-volatility estimation [31], and stochastic-volatility models [32]. These studies demonstrate the potential of physics-informed learning in computational finance. However, constructing a unified model for both Heston pricing and market calibration remains challenging. A direct concatenation of state variables and model parameters does not explicitly reflect how the parameters affect the differential operator, while a small price approximation error does not necessarily guarantee reliable parameter sensitivities. These limitations are important in the inverse problem because inaccurate parameter gradients can lead to unstable updates or parameter combinations that fit selected prices without consistently reproducing the market price surface. A unified framework must therefore represent the conditional pricing map, preserve the physical structure of the Heston equation, and provide sufficiently stable parameter sensitivities for subsequent calibration.

To address these issues, this paper proposes FiLM-MM-PINN, a parametric physics-informed neural network for Heston option pricing and market calibration, with additional constraints on local parameter sensitivities. Under a forward-normalized formulation, the model learns the Heston solution over the joint state–parameter domain by enforcing the pricing equation and its terminal and boundary conditions, without using reference option prices as training labels. A moment-matched Black–Scholes price is introduced as a structural prior, allowing the network to focus on the residual correction induced by stochastic volatility. The Heston parameters and operator-related features are processed by a dedicated parameter encoder and incorporated into the state representation through feature-wise linear modulation. Local parameter-sensitivity constraints are introduced progressively during training to improve the consistency of the learned price derivatives. Once training is completed, the network weights are frozen, and only the Heston parameter vector, including the initial variance, is optimized against an S&P 500 Index (SPX) option chain through a multi-start, staged procedure. This offline–online construction replaces repeated numerical solutions of the Heston equation during calibration with differentiable evaluations of a unified pricing model.

The principal contributions of this work are summarized as follows:

- A forward-normalized physics-informed model is developed to approximate the Heston pricing map over the joint state–parameter domain without reference prices as training labels.
- A sensitivity-regularized FiLM-MM-PINN architecture is constructed by combining a moment-matched Black–Scholes prior, operator-related parameter features, conditional feature modulation, and progressive local sensitivity constraints.
- The trained pricing model is integrated with a multi-start, staged inverse procedure for calibration to an SPX option chain and is evaluated through forward-pricing tests, sensitivity analysis, structural ablation, held-out market data, and comparisons with the Fourier–cosine expansion method and the alternating direction implicit finite-difference method.

The remainder of this paper is organized as follows. Section 2 presents the forward-normalized Heston problem, the FiLM-MM-PINN architecture, the physics-informed training strategy, and the market inverse formulation. Section 3 reports the forward-pricing, sensitivity, ablation, and market-calibration results, together with comparisons against conventional numerical methods. Section 4 presents the conclusions of this study.

2. Methodology

This paper develops a parametric physics-informed neural network framework for Heston option pricing and market calibration, termed FiLM-MM-PINN, where FiLM denotes feature-wise linear modulation and MM refers to the moment-matched Black–Scholes pricing prior. In the forward problem, the network learns the mapping

$$(y, v, \tau, \kappa, \theta, \xi, \rho) \longmapsto u(y, v, \tau; \kappa, \theta, \xi, \rho).$$

Over a prescribed joint state–parameter domain, where y denotes forward log-moneyness, v is the instantaneous variance, τ is the time to maturity, and u is the normalized European call price. The parameters κ, θ, ξ, and ρ represent the variance mean-reversion speed, long-run variance, volatility of variance, and correlation between asset-price and variance innovations, respectively. The forward model is trained by enforcing the Heston pricing equation together with its terminal and boundary conditions, without using reference option prices as training labels.

Once forward training is completed, the network weights are frozen, and the inverse problem estimates only the five-dimensional parameter vector

$$\boldsymbol{\vartheta} = (\kappa, \theta, \xi, \rho, v_0),$$

where v_0 denotes the initial variance, by minimizing the discrepancy between model-implied prices and market quotes. This offline–online decomposition replaces repeated numerical pricing during calibration with differentiable evaluations of the trained pricing model.

The overall framework is illustrated in Figure 1. The left panel shows the offline construction of the parameterized forward model. State variables and model parameters are encoded separately and coupled through feature-wise linear modulation (FiLM) [33]. A moment-matched Black–Scholes price is used as a structural prior, while the neural network learns the remaining stochastic-volatility correction. The right panel shows the online calibration stage. With the network weights fixed, preprocessed market quotes define the inverse objective, and the Heston parameters are identified through multi-start, staged optimization.

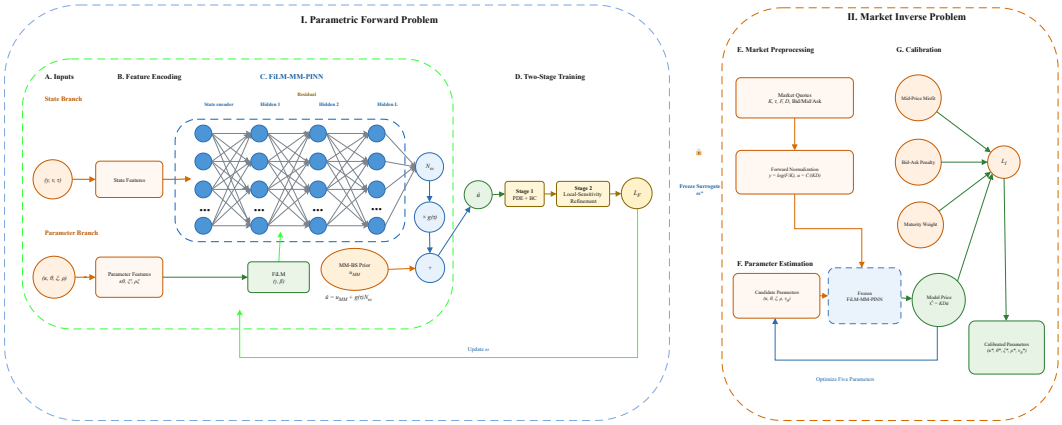


Figure 1. Unified FiLM-MM-PINN framework for parameterized Heston pricing and market calibration.

2.1. The Heston model and forward normalization

Under the risk-neutral measure, the Heston stochastic volatility model [3] assumes that the underlying asset price S_t and its instantaneous variance v_t satisfy

$$\begin{aligned} dS_t &= (r-q)S_t\,dt + \sqrt{v_t}S_t\,dW_t^S, \\ dv_t &= \kappa(\theta - v_t)\,dt + \xi\sqrt{v_t}\,dW_t^v, \\ dW_t^S\,dW_t^v &= \rho\,dt. \end{aligned}$$

Here, r and q denote the risk-free interest rate and continuous dividend yield, respectively. The parameter $\kappa > 0$ is the mean-reversion rate, $\theta > 0$ is the long-run variance, $\xi > 0$ is the volatility of variance, and $\rho \in [-1, 1]$ is the instantaneous correlation between the Brownian motions driving the asset return and variance. The structural parameter vector is

$$\boldsymbol{\mu} = (\kappa, \theta, \xi, \rho).$$

Market option contracts differ in strike, maturity, forward price, and discount factor. Direct approximation of unnormalized prices would require the network to represent both contract-dependent price scales and stochastic-volatility effects. We therefore define the forward-normalized variables

$$y = \log\frac{F(t,T)}{K}, \qquad \tau = T - t, \qquad u(y, v, \tau; \boldsymbol{\mu}) = \frac{C(S, v, t; K, T)}{KD(t, T)}, \tag{2.1}$$

where K and T are the strike and maturity, $C(S, v, t; K, T)$ is the European call price, $F(t, T)$ is the forward price, and $D(t, T)$ is the discount factor. Under constant r and q,

$$F(t,T) = S_t e^{(r-q)\tau}, \qquad D(t,T) = e^{-r\tau}.$$

The variable y is the log-forward moneyness. Positive, negative, and zero values correspond to forward in-the-money, out-of-the-money, and at-the-money options, respectively.

The normalized terminal payoff is

$$u(y, v, 0) = (e^y - 1)^+, \qquad (x)^+ = \max(x, 0).$$

Applying Eq (2.1) to the Heston pricing equation gives

$$\mathcal{R}_H[u] := u_\tau - \frac{1}{2}v\left(u_{yy} - u_y\right) - \rho\xi v\,u_{yv} - \frac{1}{2}\xi^2 v\,u_{vv} - \kappa(\theta - v)u_v = 0. \tag{2.2}$$

The strike, interest rate, and dividend yield no longer appear explicitly in Eq (2.2); their effects are absorbed into the forward price, discount factor, and normalized state variables. A single model can therefore represent normalized option prices across different contracts and admissible Heston parameter vectors.

Rather than training an independent PINN for each fixed parameter vector, we construct a parameterized pricing model $\widehat{u} = \widehat{u}(y, v, \tau; \boldsymbol{\mu}; \boldsymbol{\omega})$, where $\boldsymbol{\omega}$ denotes the trainable network parameters. The model approximates the family of Heston solutions over the joint state–parameter domain and can evaluate new state and parameter inputs without solving a separate PDE for each parameter configuration. This perspective is related to operator-learning methods, which aim to approximate families of solutions associated with parameter-dependent differential equations [34–36]. Unlike purely data-driven neural operators, however, the proposed model is trained by enforcing the normalized Heston equation together with its terminal and boundary conditions.

2.2. FiLM-MM-PINN architecture

FiLM-MM-PINN consists of a state encoder, an operator-aware parameter encoder, a FiLM-conditioned residual backbone, and a moment-matched Black–Scholes prior. The separate encoders distinguish the state dependence of the pricing surface from the parameter dependence of the Heston operator, while feature-wise linear modulation couples the two representations within the hidden layers.

2.2.1. State-parameter encoding and FiLM conditioning

The state input is $z = (y, v, \tau)$. In addition to the normalized state variables, smooth transformations of v and τ are included to improve the representation of low-variance and short-maturity regions. Interior collocation points satisfy $\tau \geq \tau_{\min} > 0$, whereas compatibility with the terminal payoff is imposed through the output structure.

The parameter encoder takes $\boldsymbol{\mu} = (\kappa, \theta, \xi, \rho)$, together with the operator-related features $\kappa\theta$, ξ^2, $\rho\xi$, $2\kappa\theta - \xi^2$. The first three composite quantities appear directly in the variance-drift, variance-diffusion, and mixed-derivative coefficients of the normalized Heston equation. The final quantity measures the margin associated with the Feller condition for the square-root variance process [37]. It is supplied to the parameter encoder as an informative feature rather than imposed as a hard constraint.

Let $\boldsymbol{h}^{(\ell)}$ denote the input hidden state at layer ℓ, and let $\boldsymbol{r}^{(\ell)}$ denote the corresponding nonlinear residual feature. Following the feature-wise linear modulation mechanism [33], the parameter encoder generates layer-dependent scaling and shifting vectors, denoted by $\boldsymbol{\gamma}^{(\ell)}(\boldsymbol{\mu})$ and $\boldsymbol{\beta}^{(\ell)}(\boldsymbol{\mu})$, respectively. The modulated residual feature is defined as

$$\widetilde{\boldsymbol{h}}^{(\ell)} = \left[\mathbf{1} + a_\gamma \tanh\left(\boldsymbol{\gamma}^{(\ell)}(\boldsymbol{\mu})\right)\right] \odot \boldsymbol{r}^{(\ell)} + a_\beta \tanh\left(\boldsymbol{\beta}^{(\ell)}(\boldsymbol{\mu})\right),$$

where $\odot$ denotes element-wise multiplication, and a_γ and a_β control the maximum amplitudes of the multiplicative and additive modulation, respectively. Compared with the standard affine FiLM transformation, the identity-centered scaling and bounded hyperbolic tangent functions limit the magnitude of parameter-induced perturbations to the hidden features. The modulated feature is then combined with the incoming hidden state through a residual connection [38]. Layer normalization and Sigmoid Linear Unit(SiLU) activation functions are applied within each residual block. The final FiLM projections are initialized to zero, so that the initial modulation is close to the identity mapping and the influence of the Heston parameters is introduced progressively during training.

2.2.2. Moment-matched Black–Scholes prior

To reduce the approximation difficulty near maturity, we construct a Black–Scholes prior using the conditional expected average variance

$$\bar{v}(\tau) = \frac{1}{\tau}\int_0^\tau \mathbb{E}[v_s \mid v_0 = v]\,\mathrm{d}s = \theta + (v - \theta)\frac{1 - e^{-\kappa\tau}}{\kappa\tau},$$

with the continuous limit $\bar{v}(0) = v$, where the variance process is initialized at the current state v and time is measured from the valuation date. The corresponding normalized Black–Scholes price is

$$u_{\mathrm{MM}} = e^y \Phi(d_1) - \Phi(d_2),$$

where Φ denotes the standard normal cumulative distribution function, the subscript MM refers to moment matching, and

$$d_1 = \frac{y + \frac{1}{2}\bar{v}(\tau)\tau}{\sqrt{\bar{v}(\tau)\tau}}, \qquad d_2 = d_1 - \sqrt{\bar{v}(\tau)\tau}.$$

The output of FiLM-MM-PINN is defined by

$$\widehat{u}(y, v, \tau; \boldsymbol{\mu}) = u_{\mathrm{MM}}(y, v, \tau; \kappa, \theta) + a_u g(\tau) N_\omega(y, v, \tau; \boldsymbol{\mu}), \tag{2.3}$$

where N_ω denotes the FiLM-conditioned correction network with trainable weights $\boldsymbol{\omega}$, a_u is an output scaling factor, and $g(\tau)$ is a maturity-dependent gating function defined by

$$g(\tau) = \left(1 - e^{-\tau/\tau_g}\right)^p, \qquad \tau_g > 0, \quad p > 0,$$

with τ_g controlling the transition scale and p controlling the shape of the gate. Since $g(0) = 0$, the neural correction vanishes at maturity, and Equation (2.3) recovers the normalized terminal payoff $(e^y - 1)^+$ through the continuous limit of the Black–Scholes prior. This construction follows the general principle of embedding essential conditions directly into neural-network trial functions [39]. Away from maturity, the correction network represents the remaining stochastic-volatility effects that are not captured by the moment-matched prior.

2.3. Physics-informed training and sensitivity regularization

FiLM-MM-PINN is trained without market prices or Fourier–cosine expansion (COS) reference values as supervised labels. The COS method is used only for independent accuracy evaluation after training.

2.3.1. Physics-informed loss

For an interior collocation point $(z_i, \boldsymbol{\mu}_i)$, where $z_i = (y_i, v_i, \tau_i)$ denotes the state vector and $\boldsymbol{\mu}_i = (\kappa_i, \theta_i, \xi_i, \rho_i)$ denotes the structural parameter vector, automatic differentiation is used to evaluate the derivatives required by Eq (2.2). Let

$$R_i = \mathcal{R}_H[\widehat{u}](z_i; \boldsymbol{\mu}_i)$$

denote the raw PDE residual, where $\mathcal{R}_H$ is the residual operator associated with Eq (2.2), and let $T_{i,m}$ denote its mth signed operator term, so that $R_i = \sum_m T_{i,m}$. To account for the local magnitudes of these terms, we introduce the scaling factor S_i and the normalized residual $\widetilde{R}_i$ as

$$S_i = 1 + \sum_m |T_{i,m}|, \qquad \widetilde{R}_i = \frac{R_i}{D_i}.$$

The normalized residual contributes only to an auxiliary term and does not replace the raw residual penalty, giving the composite PDE loss

$$\begin{aligned} \mathcal{L}_{\mathrm{pde}} &= \mathcal{L}_{\mathrm{raw}} + \lambda_{\mathrm{obr}} \mathcal{L}_{\mathrm{obr}}, \\ \mathcal{L}_{\mathrm{raw}} &= \frac{1}{N_c} \sum_{i=1}^{N_c} |R_i|^2, \\ \mathcal{L}_{\mathrm{obr}} &= \frac{1}{N_c} \sum_{i=1}^{N_c} H_{\delta_{\mathrm{pde}}}(\widetilde{R}_i), \end{aligned} \tag{2.4}$$

where N_c is the number of interior collocation points used in the current training iteration, $\mathcal{L}_{\text{raw}}$ is the mean squared raw residual, and $\mathcal{L}_{\text{obr}}$ is the auxiliary operator-balanced residual loss. The fixed, finite weight $\lambda_{\text{obr}} \geq 0$ controls the auxiliary contribution, while $\delta_{\text{pde}} > 0$ is the threshold of the Huber-type penalty; their values are specified in the experimental settings. The penalty is defined as

$$H_\delta(s) = \begin{cases} s^2, & |s| \leq \delta, \\ 2\delta|s| - \delta^2, & |s| > \delta. \end{cases}$$

During backpropagation, the denominator D_i is detached from the computational graph and treated as a pointwise scaling factor for the auxiliary term.

Since $D_i \geq 1$ and $0 \leq H_\delta(s) \leq s^2$, the auxiliary loss satisfies

$$0 \leq \mathcal{L}_{\text{obr}} \leq \frac{1}{N_c} \sum_{i=1}^{N_c} \frac{|R_i|^2}{D_i^2} \leq \mathcal{L}_{\text{raw}},$$

and consequently,

$$\mathcal{L}_{\text{raw}} \leq \mathcal{L}_{\text{pde}} \leq (1 + \lambda_{\text{obr}})\mathcal{L}_{\text{raw}}. \tag{2.5}$$

For the composite loss in Eq (2.4), the bounds in Eq (2.5) show that, on a fixed collocation set, the PDE loss tends to zero if and only if the raw residual mean squared error tends to zero, while the raw residual remains directly penalized even where the normalization denominator is large, without requiring a uniform upper bound on D_i. This establishes consistency in the residual-vanishing sense rather than equivalence of optimization trajectories or minimizers at nonzero loss, and does not by itself establish convergence over the continuous domain. Equation satisfaction is additionally assessed using the unnormalized PDE residual on an independent physics test set.

At the truncated log-moneyness boundaries $y_{\min}$ and $y_{\max}$, denoting the lower and upper limits of the computational domain in y, respectively, the following asymptotic conditions are imposed:

$$\begin{aligned} &\widehat{u}(y_{\min}, v, \tau) \approx 0, && \widehat{u}_y(y_{\min}, v, \tau) \approx 0, \\ &\widehat{u}(y_{\max}, v, \tau) \approx e^{y_{\max}} - 1, && \widehat{u}_y(y_{\max}, v, \tau) \approx e^{y_{\max}}. \end{aligned}$$

Here, subscripts on $\widehat{u}$ denote partial derivatives with respect to the indicated variables. At the variance boundaries $v = 0$ and $v = v_{\max}$, where $v_{\max}$ is the upper truncation limit of the variance domain, we impose

$$\widehat{u}_\tau - \kappa\theta\widehat{u}_v = 0 \quad \text{at } v = 0, \qquad \widehat{u}_v = 0 \quad \text{at } v = v_{\max}.$$

Collecting the value, log-moneyness derivative, and variance-boundary penalties in $\mathcal{L}_{\text{bc}}$, the base training objective is

$$\mathcal{L}_{\text{base}} = \lambda_{\text{pde}}\mathcal{L}_{\text{pde}} + \mathcal{L}_{\text{bc}},$$

where $\lambda_{\text{pde}} > 0$ controls the contribution of the PDE loss and $\mathcal{L}_{\text{bc}}$ denotes the combined boundary loss, including the weights assigned to its constituent penalties.

Collocation points are generated using Sobol low-discrepancy sequences, with the sampling distribution providing global coverage of the joint state–parameter domain while allocating additional points to short-maturity, low-variance, near-at-the-money, boundary, and parameter-extreme regions. This nonuniform allocation follows the general principle that concentrating collocation points in difficult regions can improve the training efficiency of physics-informed neural networks [40].

2.3.2. Progressive sensitivity regularization

Because the inverse problem differentiates the frozen pricing model with respect to the Heston parameters, price accuracy alone does not guarantee reliable parameter gradients. Motivated by gradient-enhanced physics-informed learning [41], the base objective is supplemented with local parameter-space constraints designed to improve the consistency of the learned pricing map across neighboring parameter configurations.

Let $\boldsymbol{s}_i$ denote the normalized structural parameter vector at the ith sampled point, with each coordinate $s_j \in [-1, 1]$, and define the neighboring parameter points by

$$\boldsymbol{s}_{i,j}^{\pm} = \boldsymbol{s}_i \pm \delta_j \boldsymbol{e}_j,$$

where $\delta_j > 0$ is the perturbation size in the normalized jth parameter coordinate and $\boldsymbol{e}_j$ is the corresponding unit vector. The state variables remain unchanged and the perturbed points are restricted to the prescribed parameter domain. Denote the raw and normalized PDE residuals at these neighboring points by $R_{i,j}^{\pm}$ and $\widetilde{R}_{i,j}^{\pm}$, respectively.

The local-pair loss enforces the Heston equation at both neighboring parameter points:

$$\mathcal{L}_{\text{pair}} = \frac{1}{2N_s} \sum_{i=1}^{N_s} \left(|R_{i,j_i}^{+}|^2 + |R_{i,j_i}^{-}|^2 \right),$$

where N_s is the number of local triplets, each consisting of a central point and its two parameter-space neighbors, and j_i denotes the parameter direction selected for the ith triplet. The variation of the normalized residual along the selected direction is constrained by

$$\mathcal{L}_{\text{tan}} = \frac{1}{N_s} \sum_{i=1}^{N_s} H_{\delta_{\text{pde}}} \left(\frac{\widetilde{R}_{i,j_i}^{+} - \widetilde{R}_{i,j_i}^{-}}{s_{i,j_i}^{+} - s_{i,j_i}^{-}} \right),$$

where $s_{i,j_i}^{\pm}$ are the selected coordinates of the neighboring parameter vectors and $H_{\delta_{\text{pde}}}$ is the Huber-type penalty defined previously. Moving estimates of the direction-specific loss magnitudes are used to balance the four parameter directions.

Additional automatic differentiation (AD) tangent constraints are applied in the ξ and ρ directions:

$$\mathcal{L}_{\text{AD},j} = \frac{1}{N_a} \sum_{i=1}^{N_a} H_{\delta_{\text{AD}}} \left(\frac{\partial \widetilde{R}_i}{\partial s_j} \right), \qquad j \in \{\xi, \rho\},$$

where N_a is the number of points used for these constraints and $\delta_{\text{AD}} > 0$ is the corresponding Huber-type penalty threshold. For a physical parameter $\mu_j \in [\mu_j^{\min}, \mu_j^{\max}]$, where $\mu_j^{\min}$ and $\mu_j^{\max}$ denote its prescribed lower and upper bounds, the normalized coordinate and its differentiation rule are

$$s_j = 2\frac{\mu_j - \mu_j^{\min}}{\mu_j^{\max} - \mu_j^{\min}} - 1, \qquad \frac{\partial}{\partial s_j} = \frac{\mu_j^{\max} - \mu_j^{\min}}{2} \frac{\partial}{\partial \mu_j}.$$

A finite-increment consistency term is further imposed in the ξ direction, with the increment defined as

$$h_\xi = \eta_\xi (\xi_{\max} - \xi_{\min}),$$

where $\eta_\xi > 0$ specifies the increment as a fraction of the prescribed ξ-interval width. The AD and centered finite difference (FD) estimates of the price sensitivity are defined as

$$D_\xi^{\mathrm{AD}}\widehat{u} = \frac{\partial \widehat{u}}{\partial \xi}, \qquad D_\xi^{\mathrm{FD}}\widehat{u} = \frac{\widehat{u}(\xi + h_\xi) - \widehat{u}(\xi - h_\xi)}{2h_\xi}.$$

The unchanged variables $(z, \kappa, \theta, \rho)$ are omitted from the two network evaluations for compactness. The normalized consistency loss is

$$\mathcal{L}_{\xi,\mathrm{AD-FD}} = \frac{1}{N_f}\sum_{i=1}^{N_f} H_{\delta_{\mathrm{FD}}}\left(\frac{D_\xi^{\mathrm{AD}}\widehat{u}_i - D_\xi^{\mathrm{FD}}\widehat{u}_i}{\varepsilon_\xi + \frac{1}{2}\left(|D_\xi^{\mathrm{AD}}\widehat{u}_i| + |D_\xi^{\mathrm{FD}}\widehat{u}_i|\right)}\right),$$

where N_f is the number of points used for the finite-increment consistency constraint, $\varepsilon_\xi > 0$ stabilizes the normalization denominator, and $\delta_{\mathrm{FD}} > 0$ is the corresponding Huber-type penalty threshold.

The complete forward-training objective is

$$\begin{aligned}\mathcal{L}_{\mathrm{forward}}(e) = \mathcal{L}_{\mathrm{base}} &+ w_{\mathrm{pair}}(e)\mathcal{L}_{\mathrm{pair}} + w_{\mathrm{tan}}(e)\mathcal{L}_{\mathrm{tan}} \\ &+ w_{\mathrm{AD},\xi}(e)\mathcal{L}_{\mathrm{AD},\xi} + w_{\mathrm{AD},\rho}(e)\mathcal{L}_{\mathrm{AD},\rho} \\ &+ w_{\xi,\mathrm{AD-FD}}(e)\mathcal{L}_{\xi,\mathrm{AD-FD}},\end{aligned} \tag{2.6}$$

where e denotes the training epoch and the nonnegative functions $w_{\mathrm{pair}}(e)$, $w_{\mathrm{tan}}(e)$, $w_{\mathrm{AD},\xi}(e)$, $w_{\mathrm{AD},\rho}(e)$, and $w_{\xi,\mathrm{AD-FD}}(e)$ specify the epoch-dependent weights of the corresponding auxiliary losses. The base loss remains active throughout training, whereas the sensitivity terms are introduced progressively by increasing their weights smoothly from zero. All loss terms are optimized along a single continuous training trajectory without resetting the network, optimizer, or learning-rate schedule, and the final network weights $\boldsymbol{\omega}^*$ are selected using the physics-only validation criterion.

2.4. Market calibration with the frozen pricing model

After forward training, the network weights $\boldsymbol{\omega}^*$ are frozen, and market calibration updates only the five-dimensional parameter vector

$$\boldsymbol{\vartheta} = (\kappa, \theta, \xi, \rho, v_0),$$

where κ, θ, ξ, and ρ enter the parameter encoder, while the initial variance v_0 enters the pricing model as a state variable. All contracts observed on the same valuation date share the same parameter vector.

For the ith market option, the strike K_i, discount factor D_i, forward price F_i, and remaining maturity τ_i are known, and its model-implied price in currency units is

$$\widehat{C}_i(\boldsymbol{\vartheta}) = K_i D_i \widehat{u}\left(\log\frac{F_i}{K_i}, v_0, \tau_i; \kappa, \theta, \xi, \rho; \boldsymbol{\omega}^*\right). \tag{2.7}$$

With $\boldsymbol{\omega}^*$ held fixed, model prices are obtained through forward evaluations of the network, while the parameter gradients required for calibration are computed by automatic differentiation with respect to $\boldsymbol{\vartheta}$.

2.4.1. Market calibration objective

Let B_i, M_i, and A_i denote the bid, mid, and ask prices of the ith option, respectively, and let $\mathrm{ReLU}(x) = \max(x, 0)$. The normalized midpoint residual is

$$r_i^{\mathrm{mid}} = \frac{\widehat{C}_i - M_i}{c_0 + |M_i|},$$

where $c_0 > 0$ is a price-scale constant that prevents excessive scaling for low-priced options. The distance from the bid–ask interval is

$$d_i^{\mathrm{BA}} = \mathrm{ReLU}(B_i - \widehat{C}_i) + \mathrm{ReLU}(\widehat{C}_i - A_i),$$

and the corresponding normalized residual is

$$r_i^{\mathrm{BA}} = \frac{d_i^{\mathrm{BA}}}{c_0 + |M_i|}.$$

Thus, no interval penalty is imposed when $\widehat{C}_i \in [B_i, A_i]$.

Let N denote the number of option quotes used in the current calibration objective and w_i the observation weight obtained from the preprocessed quote reliability measures. The weights are clipped, normalized, and balanced across maturities, and the midpoint and bid–ask losses are defined as

$$\mathcal{L}_{\mathrm{mid}} = \frac{1}{N} \sum_{i=1}^{N} H_{\delta_{\mathrm{mid}}}\left(\sqrt{w_i}\, r_i^{\mathrm{mid}} \right)$$

and

$$\mathcal{L}_{\mathrm{spread}} = \frac{1}{N} \sum_{i=1}^{N} H_{\delta_{\mathrm{spread}}}\left(\sqrt{w_i}\, r_i^{\mathrm{BA}} \right),$$

where H_δ is the Huber-type penalty defined previously, and $\delta_{\mathrm{mid}} > 0$ and $\delta_{\mathrm{spread}} > 0$ are the respective penalty thresholds.

To align calibration with the reported currency-price error, we define the weighted relative squared-error loss as

$$\mathcal{L}_{\mathrm{cur}} = \frac{\sum_{i=1}^{N} w_i (\widehat{C}_i - M_i)^2}{\sum_{i=1}^{N} w_i M_i^2}.$$

Let N_T denote the number of distinct maturities and $\mathcal{I}_m$ the set of contracts belonging to the mth maturity. The maturity-balanced loss is

$$\mathcal{L}_{\mathrm{mat}} = \frac{1}{N_T} \sum_{m=1}^{N_T} \frac{\sum_{i \in \mathcal{I}_m} w_i (\widehat{C}_i - M_i)^2}{\sum_{i \in \mathcal{I}_m} w_i M_i^2}.$$

Three parameter regularizers are included. First, the scaled prior penalty is

$$\mathcal{L}_{\mathrm{prior}} = \frac{1}{5} \sum_{j=1}^{5} \left(\frac{\vartheta_j - \vartheta_j^{\mathrm{ref}}}{\vartheta_j^{\max} - \vartheta_j^{\min}} \right)^2,$$

where ϑ_j^{ref} is the reference value of parameter j, and $\vartheta_j^{\min}$ and $\vartheta_j^{\max}$ are its prescribed lower and upper calibration bounds, respectively.

Second, a boundary penalty discourages parameters from approaching the edges of their prescribed calibration intervals. We define

$$\pi_j = \frac{\vartheta_j - \vartheta_j^{\min}}{\vartheta_j^{\max} - \vartheta_j^{\min}},$$

where π_j denotes the relative position of parameter j within its calibration interval. The penalty is

$$\mathcal{L}_{\text{bar}} = \sum_{j=1}^{5} \left[\text{ReLU}(\epsilon_b - \pi_j)^2 + \text{ReLU}(\pi_j - 1 + \epsilon_b)^2 \right],$$

where $0 < \epsilon_b < 1/2$ specifies the boundary margin over which the penalty becomes active.

Third, the soft Feller penalty is

$$\mathcal{L}_{\text{feller}} = \left[\text{ReLU}\left(\xi^2 - 2\kappa\theta\right) \right]^2.$$

The Feller condition is treated as a soft penalty rather than a hard constraint, allowing the calibration to retain parameter vectors supported by market data even when the condition is not strictly satisfied.

The inverse objective is

$$\begin{aligned} \mathcal{L}_{\text{inverse}} = {} & \lambda_{\text{mid}}\mathcal{L}_{\text{mid}} + \lambda_{\text{spread}}\mathcal{L}_{\text{spread}} + \lambda_{\text{cur}}\mathcal{L}_{\text{cur}} \\ & + \lambda_{\text{mat}}\mathcal{L}_{\text{mat}} + \lambda_{\text{prior}}\mathcal{L}_{\text{prior}} + \lambda_{\text{bar}}\mathcal{L}_{\text{bar}} + \lambda_{\text{feller}}\mathcal{L}_{\text{feller}}. \end{aligned} \tag{2.8}$$

The nonnegative coefficients λ_{mid}, λ_{spread}, λ_{cur}, λ_{mat}, λ_{prior}, λ_{bar}, and λ_{feller} control the contributions of the corresponding loss terms, with their relative values adjusted between the multi-start search and final refinement stages. The search objective emphasizes informative quotes and maturity balance, whereas the final objective places greater weight on the global currency-price relative error used for model assessment.

2.4.2. Bounded parameterization and multi-start optimization

Because FiLM-MM-PINN is trained over a bounded parameter domain, all candidate parameters are kept within that domain during calibration. An unconstrained variable η_j is mapped to the physical parameter through

$$\vartheta_j = \vartheta_j^{\min} + \left(\vartheta_j^{\max} - \vartheta_j^{\min}\right)\sigma\left(\frac{\eta_j}{T_\eta}\right), \tag{2.9}$$

where σ is the logistic sigmoid and $T_\eta > 0$ controls the scaling of the latent variable.

The Heston inverse problem is non-convex and contains compensating parameter effects. We therefore optimize progressively over the parameter groups

$$v_0 \longrightarrow (v_0, \theta) \longrightarrow (v_0, \theta, \kappa) \longrightarrow (v_0, \xi, \rho) \longrightarrow (\kappa, \theta, \xi, \rho, v_0).$$

The first stages identify the overall variance level and mean-reversion structure. The fourth stage focuses on the volatility-of-variance and correlation parameters that primarily control the smile and skew, after which all five parameters are refined jointly.

The initial candidates combine economically motivated deterministic values with Sobol multi-start samples over the bounded parameter domain. Each candidate is optimized using Adam with a stage-dependent learning rate. Candidate states are retained when their calibration loss is within a prescribed tolerance of the best training loss, and the internal selection set is used to distinguish among these candidates. After the staged Adam search, the selected candidates are refined jointly using the limited-memory Broyden–Fletcher–Goldfarb–Shanno method. The final parameter vector is selected without using the held-out validation set. That set is evaluated only once after the calibration procedure has been completed.

The calibrated Heston parameters are

$$\boldsymbol{\vartheta}^* = \arg\min_{\boldsymbol{\vartheta}} \mathcal{L}_{\text{inverse}}(\boldsymbol{\vartheta}) = (\kappa^*, \theta^*, \xi^*, \rho^*, v_0^*).$$

3. Numerical experiments

To evaluate FiLM-MM-PINN for Heston option pricing and market parameter calibration, we conduct numerical experiments covering forward-pricing accuracy, computational-configuration sensitivity, structural ablation, parameter sensitivities, boundary and out-of-domain performance, synthetic parameter recovery, SPX market calibration, and computational cost. All experiments are implemented in Python 3.10, with the principal numerical computations performed on an NVIDIA GeForce RTX 5090 GPU and double-precision floating-point arithmetic used throughout forward training, inverse calibration, and the numerical benchmarks. PyTorch is used for the neural-network computations, while GPU implementations of the COS and ADI methods provide the conventional numerical benchmarks. The Fourier–cosine expansion (COS) method [6] supplies independent reference prices for forward and sensitivity evaluation, generates synthetic option prices for parameter-recovery experiments, and serves as a benchmark for market calibration. No COS prices, sensitivities, or calibrated parameters are used as supervised labels during forward-network training; in the synthetic inverse experiments, COS-generated prices serve only as calibration observations after the network weights have been frozen.

3.1. Parametric Heston forward problem

The forward-normalized variables, Heston pricing equation, network architecture, and physics-informed constraints are presented in Section 2. FiLM-MM-PINN is trained over the following Heston parameter domain, which was progressively adjusted during preliminary numerical experiments:

$$\kappa \in [1.0, 3.0], \qquad \theta \in [0.025, 0.085], \qquad \xi \in [0.15, 1.30], \qquad \rho \in [-0.85, -0.50].$$

The corresponding state-variable training domain is

$$y \in [-3, 3], \qquad v \in [0, 1], \qquad \tau \in [10^{-4}, 1].$$

The boundary $v = 0$ is the degenerate boundary of the Heston equation. For interior collocation points, the lower bound of τ is set to a small positive value so that the PDE derivatives are not evaluated directly at the nonsmooth terminal payoff. Terminal compatibility at $\tau = 0$ is enforced separately through the output structure.

FiLM-MM-PINN contains a state backbone and a parameter encoder coupled through feature-wise linear modulation. The state backbone has five hidden layers with a width of 129, whereas the parameter encoder has two hidden layers with a width of 64. The network uses SiLU activation functions, layer normalization, and residual connections. Its output combines a moment-matched Black–Scholes prior with a maturity-gated neural correction.

Training points are generated using Sobol low-discrepancy sequences and sampled from the interior state–parameter domain and the truncated state boundaries. The interior samples combine global coverage with increased sampling in short-maturity, low-variance, near-at-the-money, boundary, and parameter-extreme regions. The PDE residual and boundary conditions remain active throughout training. Local parameter-pair and tangent-residual constraints are introduced progressively, followed by additional automatic-differentiation constraints in the ξ and ρ directions and an AD–FD consistency constraint in the ξ direction. The complete training objective is given in Eq (2.6).

Forward-pricing performance is evaluated on two independent test sets. The uniform test set covers the core regions $y \in [-1, 1]$ and $v \in [0.005, 0.5]$, while the challenging test set places greater emphasis on short-maturity, low-variance, and near-at-the-money configurations. Mean squared error and relative L^2 error are computed against independent COS reference prices. The root mean square error(RMSE) of the PDE residual is evaluated separately to measure consistency with the normalized Heston equation.

On the uniform test set, FiLM-MM-PINN achieves a relative L^2 error of 9.224×10^{-4} and a mean squared error of 3.312×10^{-7}, while the relative L^2 error on the hard test set is 7.838×10^{-4}. The unnormalized PDE residual RMSE on the independent physics test set is 1.132×10^{-3}. The uniform test set evaluates overall pricing performance under the prescribed sampling distribution, whereas the hard test set provides a complementary assessment in predefined regions of interest. Since the two sets have different sample distributions and reference-price norms, their relative errors should not be directly interpreted as measures of their comparative difficulty.

To examine the effects of collocation sampling and network size on accuracy, we use a baseline configuration with an interior collocation batch size of $N_c = 8192$, a state-network depth of $d = 5$, and a width of $w = 129$, and vary one factor at a time while retaining the remaining baseline settings. The tested values are $N_c \in \{2048, 4096, 8192\}$, $d \in \{3, 4, 5\}$, and $w \in \{64, 96, 129\}$, where N_c denotes the number of interior collocation points resampled per iteration rather than the total number of distinct points used throughout training. All configurations use seed 42, a training budget of 20000 iterations, the same parameter domain, and fixed evaluation sets. Final evaluations use the checkpoints with the best physics-validation scores, and reference prices are not used for training or checkpoint selection.

Table 1 reports the results on common test sets, with the baseline configuration shown in the first row. All PDE RMSE values are computed from the unnormalized residual on the independent physics test set. Within the tested configurations, increasing the collocation batch size and network depth reduces both pricing errors and the PDE residual, with more pronounced relative improvements in the hard-test error and PDE RMSE. In contrast, pricing errors do not decrease monotonically with network width: width 96 produces larger pricing errors than width 64, despite a lower PDE residual, whereas width 129 achieves the lowest pricing errors and PDE RMSE among the tested widths. These results indicate that increasing network size does not necessarily reduce all errors simultaneously, and that both pricing accuracy and equation consistency should be considered when assessing different

configurations.

Table 1. Pricing accuracy under different computational configurations.

N_c	d	w	Uniform Test Rel. L^2	Hard Test Rel. L^2	PDE RMSE
8192	5	129	$\mathbf{9.2239 \times 10^{-4}}$	$\mathbf{7.8376 \times 10^{-4}}$	$\mathbf{1.1323 \times 10^{-3}}$
2048	5	129	9.3133×10^{-4}	8.5160×10^{-4}	1.4171×10^{-3}
4096	5	129	9.2502×10^{-4}	8.1354×10^{-4}	1.2725×10^{-3}
8192	3	129	9.7359×10^{-4}	1.0454×10^{-3}	1.4791×10^{-3}
8192	4	129	9.5351×10^{-4}	8.7764×10^{-4}	1.2751×10^{-3}
8192	5	64	9.5806×10^{-4}	1.0490×10^{-3}	1.6582×10^{-3}
8192	5	96	1.0192×10^{-3}	1.1815×10^{-3}	1.4728×10^{-3}

To examine error evolution during training, Table 2 reports evaluations at iterations 5000, 10000, 15000, and 20000 along the same baseline training trajectory. The pricing error is computed on a fixed monitoring set, while the unnormalized PDE residual is evaluated on a separate fixed physics-validation set. Under the adopted training schedule, both metrics decrease at the reported iterations, although the reduction in pricing error becomes smaller during later training while the PDE residual continues to decrease. This indicates continued improvement in equation consistency alongside diminishing gains in pricing accuracy. The monitoring and physics-validation sets differ from the final test sets used in Table 1; the two tables therefore describe training progress and final test performance, respectively.

Table 2. Accuracy at selected training iterations.

Iteration	Monitor Rel. L^2	Validation PDE RMSE
5000	1.1624×10^{-3}	1.8328×10^{-3}
10000	9.5044×10^{-4}	9.3237×10^{-4}
15000	9.3017×10^{-4}	7.1896×10^{-4}
20000	$\mathbf{9.2611 \times 10^{-4}}$	$\mathbf{6.5141 \times 10^{-4}}$

These experiments characterize the dependence of model accuracy on collocation batch size, network depth, network width, and training iterations within the tested settings. Larger collocation batches and deeper state networks yield lower pricing errors and PDE residuals, whereas the effect of increasing width on pricing accuracy is nonmonotonic. As training progresses, gains in pricing accuracy become smaller while the PDE residual continues to improve. These findings provide numerical support for the baseline configuration and training budget adopted in this study.

The PDE residual RMSE measures equation satisfaction but is not a direct bound on pricing or implied-volatility errors. For fixed Heston parameters, the pricing error satisfies a linear error equation driven by the residual and is also affected by terminal and boundary errors. A quantitative residual-to-price bound therefore requires an appropriate stability estimate. As v approaches zero, the second-order terms of the Heston operator degenerate, while approaching expiry introduces reduced regularity near the payoff kink. We therefore supplement the global residual metric with direct evaluations in these regions.

Two variance groups are considered: a core group with v sampled uniformly over $[0.005, 0.50]$ and a low-variance group with v sampled log-uniformly over $[10^{-6}, 10^{-2}]$. For each group, τ is sampled

log-uniformly within three intervals: $[10^{-4}, 10^{-3}]$, $[10^{-3}, 10^{-2}]$, and $[10^{-2}, 0.05]$. Near-at-the-money points are generated within

$$|y| \le 1.5\sqrt{I(v,\tau;\kappa,\theta)}, \qquad I(v,\tau;\kappa,\theta) = \theta\tau + \frac{v-\theta}{\kappa}\left(1 - e^{-\kappa\tau}\right),$$

where I is the conditional expected integrated variance. The four structural parameters are sampled within their training ranges, and each test group contains 2048 points.

Table 3 reports relative L^2 errors of normalized prices against independent COS prices and absolute implied-volatility RMSE. Pricing errors use all 2048 points per group, whereas implied volatility (IV) errors are computed only on the common eligible subset. Eligibility requires the COS reference to pass the refinement check and the COS, MM, and PINN prices to lie strictly within the normalized no-arbitrage bounds, with IV estimates within the numerical search interval $[10^{-8}, 5]$ and normalized Black–Scholes vega of at least 10^{-5}. In the last group, 44 points fail to jointly satisfy these conditions, leaving 2004 points for IV evaluation; all points are retained in the first five groups. Ineligible prices are not clipped. IV RMSE is expressed in decimal volatility units.

Table 3. Pricing and IV Errors for Short-Maturity Options Near the Money.

Variance group	Maturity interval	Price Rel. L^2	IV RMSE	IV points
Core	$[10^{-4}, 10^{-3}]$	1.909×10^{-4}	3.030×10^{-4}	2048
Core	$[10^{-3}, 10^{-2}]$	2.040×10^{-4}	3.327×10^{-4}	2048
Core	$[10^{-2}, 0.05]$	4.313×10^{-4}	6.177×10^{-4}	2048
Low	$[10^{-4}, 10^{-3}]$	3.534×10^{-2}	2.516×10^{-3}	2048
Low	$[10^{-3}, 10^{-2}]$	9.730×10^{-2}	7.320×10^{-3}	2048
Low	$[10^{-2}, 0.05]$	8.965×10^{-2}	1.026×10^{-2}	2004

The model maintains low pricing and IV errors in the tested core-variance groups, whereas larger errors occur when low variance and short maturity are combined. The corresponding MM-prior IV RMSE values in the low-variance groups are 1.798×10^{-3}, 4.610×10^{-3}, and 9.314×10^{-3}, all below the PINN values. Locally, the first-order relation

$$\Delta\sigma_{\mathrm{IV}} \approx \frac{\widehat{u} - u_{\mathrm{COS}}}{\mathrm{Vega}_u}$$

shows how pricing errors can be amplified when Vega_u, the normalized Black–Scholes vega evaluated at the reference IV, is small. This relation is a local approximation rather than a global error bound. The tests identify a challenging regime for the learned correction and demonstrate the need to assess regional pricing and IV errors alongside the global PDE residual.

We next assess the model's response to changes in the Heston parameters by comparing sensitivities of the normalized price with respect to the mean-reversion rate κ, long-run variance θ, volatility of variance ξ, and correlation ρ. Network sensitivities are obtained by automatic differentiation and compared at the same evaluation points with reference sensitivities computed by finite differences of independent COS prices. For each parameter $p \in \{\kappa, \theta, \xi, \rho\}$, the relative sensitivity error is defined as

$$E_p = \frac{\|g_{\mathrm{PINN},p} - g_{\mathrm{COS},p}\|_2}{\|g_{\mathrm{COS},p}\|_2},$$

where $g_{\mathrm{PINN},p} = \partial u_{\mathrm{PINN}}/\partial p$ and $g_{\mathrm{COS},p}$ denotes the corresponding COS reference sensitivity. The relative errors for κ, θ, ξ, and ρ are 7.6540×10^{-2}, 5.3493×10^{-2}, 8.9975×10^{-2}, and 1.2740×10^{-1}, respectively. The ξ and ρ sensitivities exhibit larger relative errors than those for κ and θ, with the largest error occurring in the ρ direction. These results indicate that low pricing errors do not necessarily imply equally accurate parameter derivatives. The COS reference sensitivities are used only for evaluation and do not participate in training.

To quantify the contribution of the neural correction, we compare the moment-matched Black–Scholes prior alone with the full FiLM-MM-PINN model at identical evaluation points. Here, $\varepsilon_{\mathrm{BS}}$ denotes the relative L^2 pricing error of the moment-matched prior, rather than a conventional Black–Scholes model with a fixed volatility, and $\varepsilon_{\mathrm{BS+PINN}}$ denotes the error after adding the learned correction. Both errors are measured against independent COS prices. The relative error reduction is calculated as

$$\frac{\varepsilon_{\mathrm{BS}} - \varepsilon_{\mathrm{BS+PINN}}}{\varepsilon_{\mathrm{BS}}} \times 100\%.$$

The results are obtained from the additional evaluation sets, each containing 2048 points, rather than the main test sets reported above.

As shown in Table 4, the MM prior provides an informative pricing baseline, while the neural correction further reduces the remaining approximation error on both the uniform and hard evaluation sets. To examine how this correction is learned under the physics-informed constraints, Figure 2 presents the evolution of the training-loss components.

Table 4. Contribution of the neural correction.

Test set	$\varepsilon_{\mathrm{BS}}$	$\varepsilon_{\mathrm{BS+PINN}}$	Reduction
Uniform	1.2796×10^{-2}	9.0666×10^{-4}	92.91%
Hard	1.2580×10^{-2}	7.1134×10^{-4}	94.35%

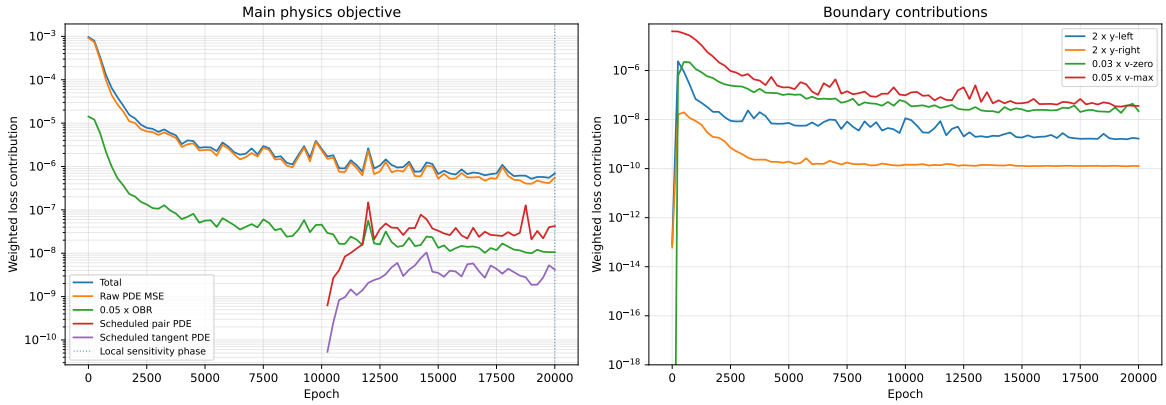
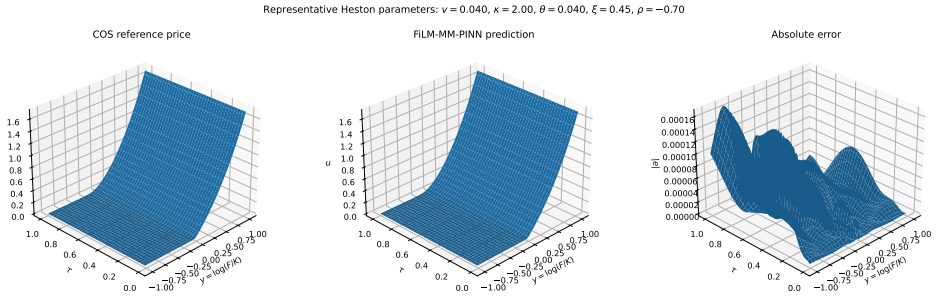


Figure 2. Evolution of the training-loss components for FiLM-MM-PINN.

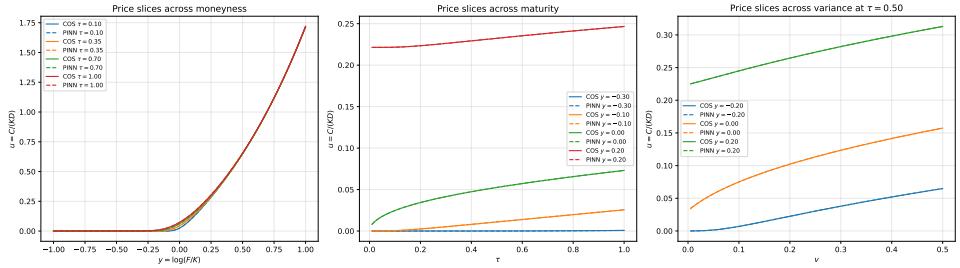
The total loss, PDE residual loss, and boundary loss decrease rapidly during the early training stage, indicating that the network progressively reduces violations of the Heston equation and the prescribed boundary conditions as it learns the price correction. The main loss components then decrease more gradually and approach stable levels. When the local parameter-pair and tangent

constraints are progressively activated in the later stage, their contributions remain controlled and do not destabilize the optimization. Overall, the decrease in the PDE and boundary losses indicates improved satisfaction of the physics-informed constraints at the sampled training points, complementing the independent test results reported above.

Following the analysis of the training-loss evolution, we examine the pricing predictions of the trained model in Figure 3, which presents the price surface and selected state-variable slices for a representative Heston parameter vector. Figure 3(a) compares the COS reference surface with the FiLM-MM-PINN prediction and shows the corresponding absolute-error distribution. The predicted surface agrees closely with the reference solution in both shape and magnitude, with the larger errors mainly occurring in regions of relatively high price curvature. Figure 3(b) compares price slices along the log-forward moneyness, time-to-maturity, and instantaneous-variance directions. The FiLM-MM-PINN predictions closely follow the COS reference curves across most of the state domain, indicating that the model captures both the global pricing surface and its local variation with respect to the state variables.



(a) References for COS prices, PINN forecast prices, and absolute error surfaces under a set of representative Heston parameters.



(b) Comparison of price profiles across different levels of value, remaining maturities, and variance.

Figure 3. A comparison of the parameterized pricing model with the COS reference solution under a set of representative Heston parameters.

To further assess generalization across different parameter configurations, Figure 4 presents the pricing results for a baseline scenario, a high volatility-of-variance scenario, and a scenario with a small or negative Feller margin. In all three cases, the FiLM-MM-PINN predictions reproduce the main spatial structure of the COS reference surfaces. The larger absolute errors are mainly concentrated in short-maturity and near-at-the-money regions, where the price surface exhibits stronger local curvature. These results indicate that the model maintains consistent forward-pricing accuracy across the tested Heston parameter configurations with different stochastic-volatility characteristics.

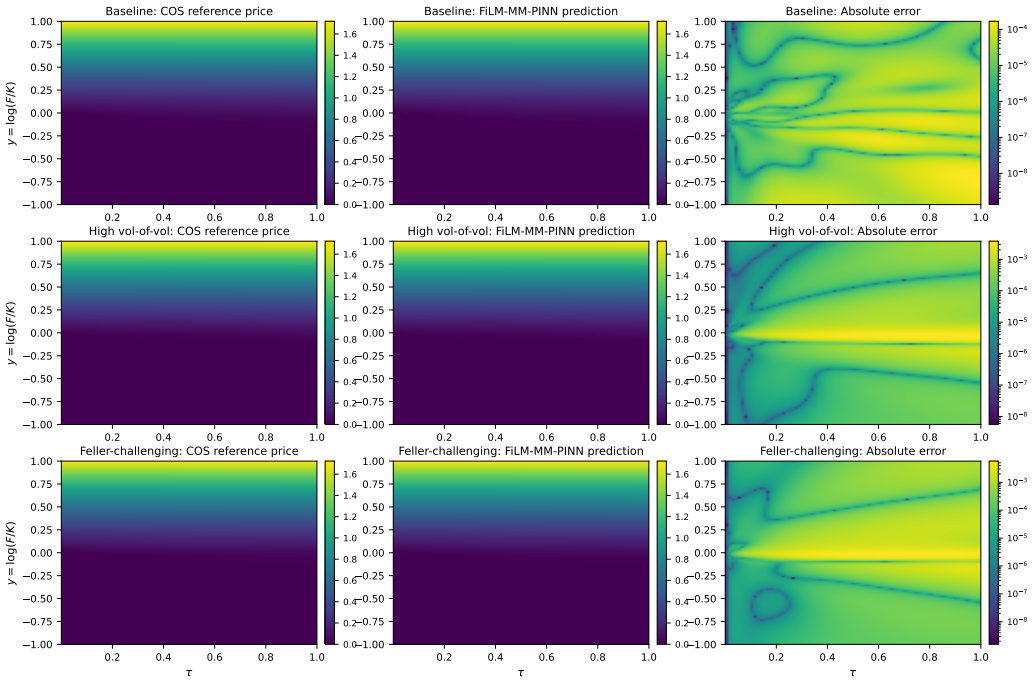


Figure 4. COS reference prices, PINN predicted prices, and the distribution of absolute errors under different parameter scenarios.

To further examine how the learned pricing map responds to individual parameter changes, we vary one Heston parameter at a time while holding the state variables and the remaining parameters fixed, obtaining the price slices shown in Figure 5. For the illustrated configuration, FiLM-MM-PINN captures the main trends of increasing prices with κ, θ, and ρ, and decreasing prices with ξ. FiLM-MM-PINN captures the increasing price trends in the κ, θ, and ρ directions and the decreasing trend in the ξ direction for the illustrated configuration. The predicted and COS reference curves nearly overlap in the θ direction, whereas differences in slope are visible in the κ and ρ directions. In the ξ direction, the discrepancy becomes more pronounced as ξ increases. These slices complement the quantitative sensitivity assessment by illustrating the price response at a selected fixed configuration, rather than across the full evaluation domain.

To assess performance near and outside the training parameter bounds, we additionally evaluate the frozen model on test groups containing 2048 points each. For a parameter with training interval $[a_p, b_p]$ and width $W_p = b_p - a_p$, the near-boundary groups use $[a_p, a_p + 0.05W_p]$ and $[b_p - 0.05W_p, b_p]$, while the neighboring out-of-domain groups use $[a_p - 0.05W_p, a_p - 0.005W_p]$ and $[b_p + 0.005W_p, b_p + 0.05W_p]$. Common Sobol coordinates are used across these groups, with only the parameter under examination changed and the remaining parameters and state variables kept identical at corresponding points. Positive correlations are separately tested over $\rho \in [0.05, 0.30]$. All errors are relative L^2 pricing errors against independent COS prices. The additional uniform in-domain group has an error of 9.0666×10^{-4} and uses different evaluation points from the main uniform test set.

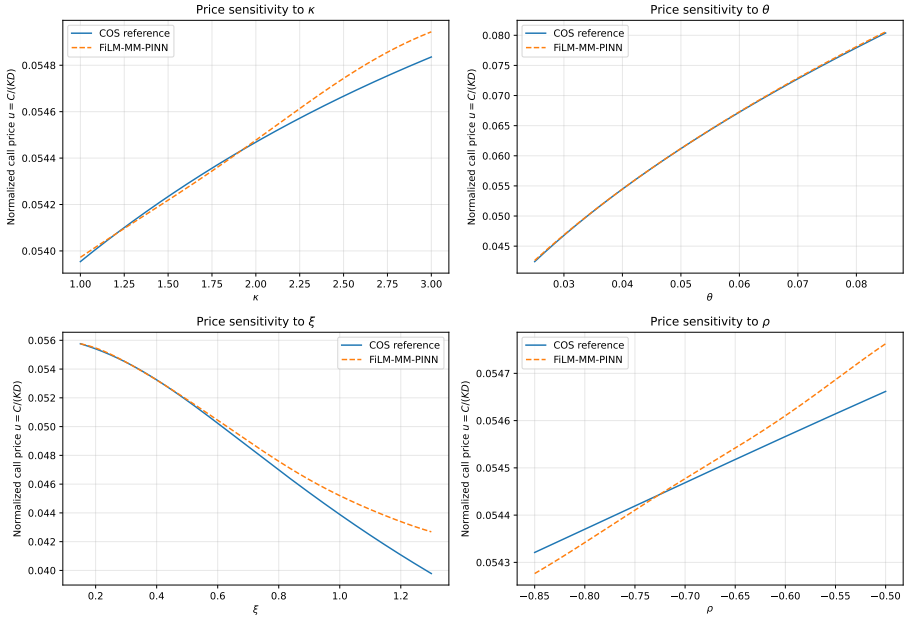


Figure 5. Normalized call-price slices with respect to the Heston parameters.

As shown in Table 5, crossing a training boundary does not uniformly increase the error. Error increases are observed beyond the lower κ bound and the upper ξ bound, whereas changes in the tested extensions in θ and around the negative-correlation bounds are smaller. A more pronounced deterioration occurs for $\rho \in [0.05, 0.30]$, where the relative L^2 error is 7.0724×10^{-3}, compared with 4.7612×10^{-3} for the MM prior. These results indicate parameter-dependent extrapolation behavior, and the accuracy established within the training domain should not be directly extended to the untrained positive-correlation region.

Table 5. Boundary and out-of-domain pricing errors.

	Near boundary		Out of domain	
Parameter	Lower	Upper	Lower	Upper
κ	1.6137×10^{-3}	4.0397×10^{-4}	1.7362×10^{-3}	3.9119×10^{-4}
θ	9.0440×10^{-4}	9.4321×10^{-4}	9.2093×10^{-4}	9.5128×10^{-4}
ξ	1.9420×10^{-4}	2.3009×10^{-3}	5.3147×10^{-4}	2.6083×10^{-3}
ρ	9.5705×10^{-4}	8.9585×10^{-4}	1.0088×10^{-3}	8.7936×10^{-4}

Structural ablation experiments are conducted to assess the contributions of the moment-matched pricing prior, operator-related parameter features, feature-wise linear modulation, and the two-branch architecture. Except for the component removed or replaced in each variant, all models use the same parameter domain, collocation-point sampling strategy, physics-informed loss, and training budget. The comparison includes the complete FiLM-MM-PINN, a model without the moment-matched prior, a model without operator-related features, a model without FiLM conditioning, and a single-branch PINN. Neither COS prices nor market observations are used as supervised training data, and COS prices are used only for independent evaluation after training.

As shown in Table 6, the complete FiLM-MM-PINN achieves the lowest pricing errors on both

test sets and the lowest PDE residual RMSE on the independent physics test set. Removing the moment-matched prior increases the pricing errors, particularly on the hard test set, while removing the operator-related features leads to a smaller deterioration across the reported metrics. The strongest deterioration occurs when FiLM conditioning is removed, indicating the importance of layer-wise parameter modulation for the tested parametric pricing task. Although the capacity-matched single-branch PINN achieves pricing errors close to those of the complete model, its larger PDE residual suggests that separating state and parameter representations and coupling them through FiLM improves equation consistency under the adopted training settings.

Table 6. Structural ablation results.

Model	Uniform Test Rel. L^2	Hard Test Rel. L^2	PDE RMSE
FiLM-MM-PINN	$\mathbf{9.224 \times 10^{-4}}$	$\mathbf{7.838 \times 10^{-4}}$	$\mathbf{1.132 \times 10^{-3}}$
Removing moment-matched prior	1.380×10^{-3}	2.268×10^{-3}	2.109×10^{-3}
Removing operator-related features	9.640×10^{-4}	1.015×10^{-3}	1.179×10^{-3}
Removing FiLM conditioning	6.937×10^{-3}	9.282×10^{-3}	1.420×10^{-2}
Single-branch PINN	9.413×10^{-4}	7.866×10^{-4}	2.894×10^{-3}

From a financial perspective, the moment-matched Black–Scholes prior provides a baseline valuation based on expected integrated variance, allowing the neural correction to focus on the remaining effects of stochastic variance and its correlation with asset returns. Operator-related features expose parameter combinations entering the pricing equation, while FiLM adjusts the state representation to different Heston parameter configurations, helping the network represent changes in prices associated with mean reversion, long-run variance, volatility of variance, and correlation. The single-branch comparison further illustrates that similar aggregate pricing errors can coexist with different levels of equation consistency, so the benefit of the proposed architecture should be assessed through both price approximation and satisfaction of the Heston equation.

3.2. Market calibration results

3.2.1. Market data and preprocessing

We use an SPX option chain obtained from the Cboe Delayed Quotes service (`https://www.cboe.com/delayed_quotes/spx/quote_table`) to evaluate the proposed method for real-market parameter calibration. The raw dataset contains call and put quotes across multiple strikes and maturities at a single observation time, together with bid and ask prices, trading volume, and open interest. The quote date and SPX spot level are extracted from the accompanying metadata to ensure that all contracts belong to the same market cross-section.

Market quotes may contain wide bid–ask spreads, low-liquidity contracts, and anomalous observations. We first compute the mid-price of each contract and retain options with times to maturity between 14 and 365 days. The contracts are then filtered using the following criteria: the relative bid–ask spread must not exceed 0.20, open interest must be at least 20, trading volume must be at least 1, and the mid-price must be at least 0.10. The sample is further restricted to

$$\left|\log(F/K)\right| \leq 0.16,$$

so that the market inputs remain within the state region in which the forward pricing model has been independently validated.

For each maturity, call and put options with the same strike are paired, and the put–call parity relation

$$C - P = D(F - K)$$

is used to estimate the maturity-specific forward price F and discount factor D. To reduce the influence of anomalous quotes, the parity relation is estimated using Huber robust regression with bid–ask-spread-based weights. Contracts with narrower spreads receive greater weights, whereas observations with large parity deviations are downweighted.

After estimating F and D for each maturity, out-of-the-money options are used to construct equivalent call quotes. When $K \geq F$, the out-of-the-money call quote is used directly. When $K < F$, the out-of-the-money put quote is converted into an equivalent call quote using put–call parity:

$$C^{\mathrm{eq}} = P + D(F - K).$$

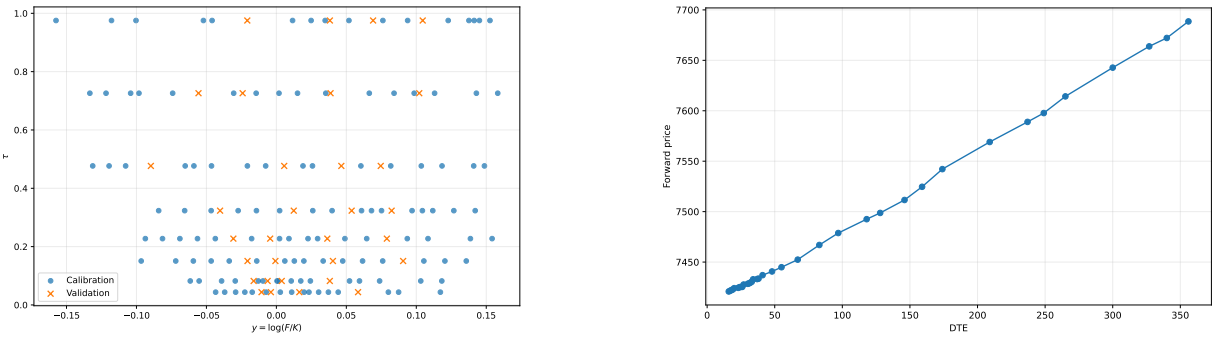
This procedure maps option quotes across different moneyness levels into a common call-price representation while retaining the more liquid out-of-the-money side of the option chain.

The resulting market observations are transformed using the same forward normalization as the pricing model:

$$y_i = \log \frac{F_i}{K_i}, \qquad u_i^{\mathrm{mkt}} = \frac{C_i^{\mathrm{mkt}}}{K_i D_i}.$$

The normalized market inputs are therefore consistent with the state variables and price scale used during forward training.

Figure 6 shows the state distribution of the preprocessed market sample and the maturity-dependent forward prices estimated from put–call parity.



(a) Filter the distribution of maturities and log-value levels for out-of-the-money option samples.

(b) The implied forward price term structure estimated from the bull–bear parity relationship.

Figure 6. Key market characteristics of the SPX options sample after preprocessing.

As shown in Figure 6(a), the filtered sample covers multiple maturities and is concentrated around the at-the-money region. It therefore provides cross-strike and cross-maturity information for the joint identification of the Heston parameters. Figure 6(b) presents the maturity-dependent forward prices implied by put–call parity. Estimating the forward price and discount factor separately for each

maturity accounts for differences in discounting and carry across the term structure and reduces their influence on the calibration of the stochastic-volatility parameters.

After preprocessing, the contracts within each maturity are stratified by moneyness and divided into a calibration set, an internal selection set, and a held-out validation set. The calibration set is used to update the Heston parameters, while the internal selection set is used to compare the candidate solutions generated by the multi-start procedure. The held-out validation set is evaluated only after the final parameter vector has been fixed. It is not used for parameter updates, initial-point screening, candidate selection, or any other model-selection decision.

3.2.2. Market calibration and benchmark results

To evaluate parameter recovery, we construct five synthetic cases, each containing 66 noise-free COS-generated option prices at 11 equally spaced log-moneyness values over $[-0.2, 0.2]$ and six maturities $\tau \in \{0.04, 0.08, 0.16, 0.32, 0.60, 1.0\}$, with $F = 100$ and $D = 1$. The frozen network is calibrated to these prices using initial points generated independently of the known generating parameters. Ordered as $(\kappa, \theta, \xi, \rho, v_0)$, the generating parameter vectors are

$$\begin{aligned}
\boldsymbol{\vartheta}^{(1)} &= (2.0, 0.040, 0.4490, -0.6995, 0.0397),\\
\boldsymbol{\vartheta}^{(2)} &= (1.4, 0.064, 0.8975, -0.7800, 0.0650),\\
\boldsymbol{\vartheta}^{(3)} &= (2.6, 0.046, 0.6675, -0.5700, 0.0298),\\
\boldsymbol{\vartheta}^{(4)} &= (1.7, 0.073, 1.1275, -0.7100, 0.0892),\\
\boldsymbol{\vartheta}^{(5)} &= (2.5, 0.034, 0.3225, -0.6225, 0.0485).
\end{aligned}$$

Parameter recovery is measured using the Euclidean error and the range-normalized parameter RMSE,

$$E_{\mathrm{par}} = \|\widehat{\boldsymbol{\vartheta}} - \boldsymbol{\vartheta}_{\mathrm{true}}\|_2, \qquad E_{\mathrm{norm}} = \sqrt{\frac{1}{5}\sum_{j=1}^{5}\left(\frac{\widehat{\vartheta}_j - \vartheta_{\mathrm{true},j}}{b_j - a_j}\right)^2},$$

where $[a_j, b_j]$ denotes the search interval of parameter j, with interval widths of 2, 0.06, 1.15, 0.35, and 0.11 for κ, θ, ξ, ρ, and v_0, respectively. We additionally report the network-based fitting error and the COS repricing error at the recovered parameters, both evaluated against the same synthetic target prices.

Table 7 summarizes the parameter-recovery results across the five synthetic cases, with network-based price-fitting errors ranging from 1.970×10^{-4} to 2.139×10^{-3} and range-normalized parameter RMSEs ranging from 1.2372×10^{-2} to 4.7063×10^{-2}. These results demonstrate the applicability of the frozen pricing network to parameter estimation across the tested Heston configurations without retraining, while also showing that recovery accuracy depends on the generating parameter vector. The COS repricing errors at the recovered parameters are consistently higher than the network-based fitting errors, suggesting that parameter adjustments may partly compensate for the network's pricing approximation error. The synthetic experiments therefore quantify both price-fitting and parameter-recovery performance, providing a controlled assessment of the inverse procedure before its application to market observations.

Table 7. Synthetic parameter recovery.

	Parameter error		Price Rel. L^2	
Case	E_{par}	E_{norm}	PINN fit	COS repricing
1	7.4700×10^{-2}	1.6905×10^{-2}	1.970×10^{-4}	3.743×10^{-4}
2	6.6377×10^{-2}	2.2001×10^{-2}	2.139×10^{-3}	6.839×10^{-3}
3	1.3246×10^{-2}	1.2372×10^{-2}	4.576×10^{-4}	8.419×10^{-4}
4	1.6980×10^{-1}	4.7063×10^{-2}	1.828×10^{-3}	5.051×10^{-3}
5	7.8710×10^{-2}	2.4184×10^{-2}	2.754×10^{-4}	9.223×10^{-4}

Following the synthetic parameter-recovery experiments, we apply the frozen pricing network to the preprocessed SPX market sample and optimize only the Heston parameters and initial variance through the multi-start, staged calibration procedure. The candidate initialized from the 23rd starting point is selected according to the calibration and internal-selection criteria, yielding

$$\kappa^* = 1.9709, \qquad \theta^* = 0.0713, \qquad \xi^* = 1.1484, \qquad \rho^* = -0.7528, \qquad v_0^* = 0.0246.$$

These estimates describe a mean-reverting variance process with substantial volatility of variance and a pronounced negative correlation between asset returns and variance innovations, consistent with the leverage effect represented by the Heston model. The calibrated long-run and initial volatility levels are approximately $\sqrt{\theta^*} = 26.7\%$ and $\sqrt{v_0^*} = 15.7\%$, respectively. The calibrated parameter combination satisfies

$$2\kappa^*\theta^* - (\xi^*)^2 < 0,$$

and therefore does not satisfy the Feller condition ensuring strict positivity of the variance process. Since this condition is imposed through a soft penalty, calibration permits a trade-off between market-price fitting and satisfaction of the condition, with the penalty weight controlling its influence on parameter estimation. The continuous-time variance process remains nonnegative, although the zero-variance boundary becomes attainable; the associated degeneracy of the pricing equation is treated through the boundary condition at $v = 0$.

To assess the pricing performance of the calibrated parameter vector, model prices are compared with market mid-quotes on the calibration and held-out validation sets, yielding a relative L^2 error of 0.2665% and an R^2 of 0.999984 on the calibration set, compared with 0.3278% and 0.999968 on the validation set. The validation error is higher but remains at the same order of magnitude, indicating that the calibrated parameters capture the main pricing structure of the observed market cross-section and retain comparable price-fitting accuracy for contracts not used in parameter estimation or candidate selection.

Figure 7 compares the model-implied prices with the observed market mid-prices. Most observations lie close to the reference line representing equality between the model and market prices, indicating consistent price fitting across the strikes and maturities in the market sample. The low relative L^2 errors and the R^2 values close to unity on both the calibration and held-out validation sets further demonstrate that the frozen FiLM-MM-PINN model can be effectively used for Heston parameter calibration.

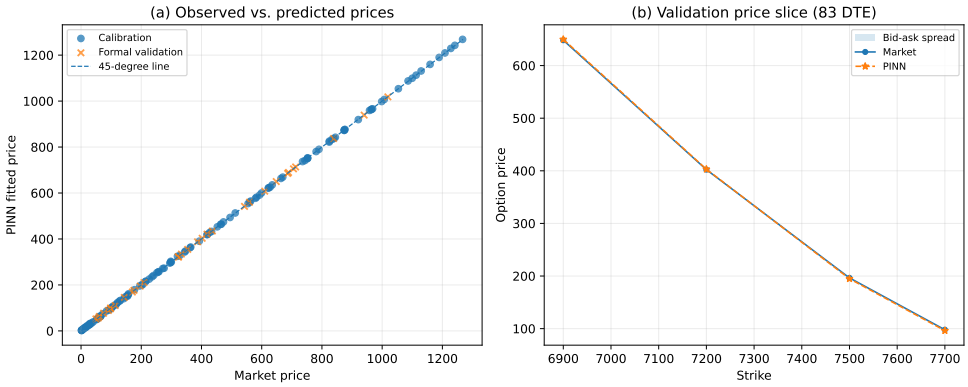


Figure 7. Results of the fit between the SPX market option mid-price and the prices predicted by the parametric Heston pricing model.

To examine the local pricing errors on the held-out sample, Figure 8 presents the price residuals, pointwise relative errors, maturity-specific relative L^2 errors, and implied-volatility fits on the validation set. For the 83-day maturity, the residuals change sign across strikes but remain small in magnitude. The pointwise relative errors are larger for some deep in-the-money or deep out-of-the-money contracts, partly because normalization by a small market price amplifies the relative error. The maturity-specific errors vary across expiration dates but show no systematic increase with the time to maturity. The model-implied volatility curves also closely track the corresponding market curves across strikes. These results indicate that the calibrated parameter vector captures both the overall option-price level and the main strike-dependent volatility structure of the held-out market sample.

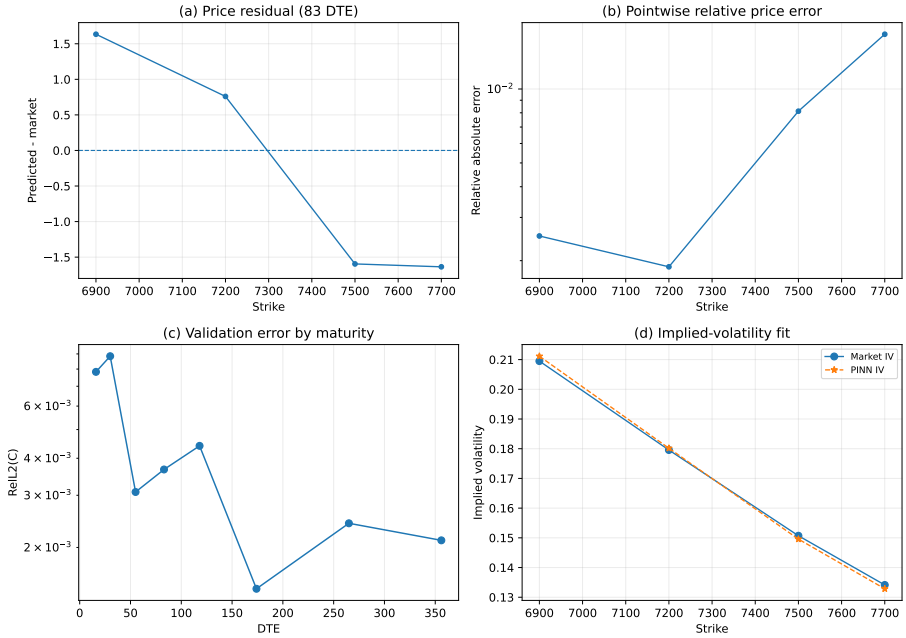


Figure 8. Price residuals, relative errors, maturity-wise errors, and implied-volatility fit on the held-out validation set.

To evaluate the performance of FiLM-MM-PINN in market calibration, we compare it with the Fourier–cosine expansion (COS) method and an alternating direction implicit (ADI) finite-difference method. All three methods use the same market samples, parameter bounds, calibration objective, and candidate-selection criteria. During calibration, the COS and ADI benchmarks repeatedly solve the pricing problem as the Heston parameters are updated, whereas FiLM-MM-PINN directly evaluates the frozen parameterized pricing network.

As shown in Table 8, FiLM-MM-PINN achieves the lowest relative L^2 errors on both the calibration and held-out validation sets, together with the highest validation R^2, indicating the closest overall agreement with market mid-prices among the three methods under the adopted calibration protocol. Whereas COS and ADI perform numerical pricing at successive candidate parameter vectors, FiLM-MM-PINN learns a parametric pricing map through offline physics-informed training and evaluates the frozen network during online calibration, providing a differentiable pricing model for multi-start, staged optimization without retraining its weights.

Table 8. Comparison of SPX market inversion results using different pricing methods.

Method	Calibration Set Rel. L^2	Validation Set Rel. L^2	Validation Set R^2
FiLM-MM-PINN	**0.2665%**	**0.3278%**	**0.999968**
COS Method	0.3968%	0.5212%	0.999919
ADI Finite Difference Method	0.4329%	0.5624%	0.999906

For the ADI finite-difference benchmark used in the market-fitting comparison, the recorded single-run time for the complete calibration is 13730.224 s, reflecting the substantial computational cost of repeated PDE solutions during start screening, Powell optimization, and least-squares refinement. Table 9 further compares the computational costs of FiLM-MM-PINN and COS using the same market samples, parameter bounds, calibration objective, multi-start staged-optimization procedure, and candidate-selection criteria. Parameter derivatives are obtained through automatic differentiation for FiLM-MM-PINN and finite differences for the COS implementation, while online timings cover all calibration starts and subsequent refinement stages but exclude model loading and warm-up, final diagnostics, and result output; both methods are run five times with GPU synchronization at the timing boundaries, and the elapsed times are summarized by their median and range.

Table 9. Computational costs of market calibration.

Method	Training (s)	Online median (s)	Online range (s)
FiLM-MM-PINN	5018.608	719.507	716.300–723.772
COS	—	1443.269	1439.558–1446.462

As shown in Table 9, FiLM-MM-PINN requires 5018.608 s of offline training and subsequently achieves a median online calibration time of 719.507 s, compared with 1443.269 s for the specified COS implementation under the same staged calibration procedure, corresponding to an approximately 2.01-fold online speedup. This reduction in online cost is accompanied by an initial investment in network training, which can be amortized through reuse of the frozen pricing map across subsequent calibration tasks within the prescribed state–parameter domain, provided that no retraining is required.

4. Conclusions

This paper proposes FiLM-MM-PINN, a sensitivity-regularized parametric solver for Heston option pricing and market calibration, which learns the solution over the joint state–parameter domain by enforcing the forward-normalized Heston PDE together with its terminal and boundary conditions. A moment-matched Black–Scholes prior provides the baseline pricing structure, while operator-related parameter features and feature-wise linear modulation capture the dependence of the solution on the Heston parameters, with local sensitivity constraints introduced to improve the consistency of parameter derivatives. After forward training, the network weights are frozen and embedded in a multi-start, staged procedure for calibration to an SPX option chain, thereby separating physics-informed forward training from market parameter identification.

Numerical results show that FiLM-MM-PINN accurately approximates Heston price surfaces within the prescribed state–parameter domain, while ablation results support the contributions of the moment-matched prior, operator-related features, and FiLM conditioning to pricing accuracy and equation consistency. On the held-out market set, the calibrated model achieves a relative L^2 error of 0.3278% and reproduces the main implied-volatility structure across strikes, with lower price-fitting errors than the calibrated COS and ADI benchmarks under the adopted calibration protocol. These results support the use of a frozen differentiable pricing model for Heston calibration, although the additional evaluations show that low aggregate pricing errors do not ensure uniformly accurate local predictions, extrapolation, or parameter recovery. Improving these aspects motivates further work on targeted sampling and parameter-sensitivity accuracy, together with an examination of the soft Feller penalty's influence on calibration and extensions to broader parameter domains and more complex stochastic-volatility models.

Author contributions

Jina Li: Conceptualization, Methodology, Funding acquisition, Supervision, Writing–review and editing; Daiyuan Jia: Software, Validation, Formal analysis, Investigation, Writing original draft, Visualization; Yanpei Zhen: Supervision, Writing–review and editing; Yaxin Li: Writing–review and editing, Validation. All authors have read and approved the final version of the manuscript for publication.

Use of Generative-AI tools declaration

The authors confirm that AI tools (e.g., ChatGPT and Doubao) were used solely for polishing the English language and improving the readability of the manuscript. The authors take full responsibility for the accuracy and originality of the entire content. No AI was used to generate data, results, or core scientific conclusions.

Data availability statement

The market data were obtained from the Cboe Delayed Quotes service (`https://www.cboe.com/delayed_quotes/spx/quote_table`) and are subject to its terms of

use. Processed data and implementation details are available from the corresponding author upon reasonable request, subject to these terms.

Acknowledgments

This work was supported by the Natural Science Foundation of Henan Province (Grant No. 252300421500) and the Advantageous Disciplines Strength Improvement Project of Zhongyuan University of Technology (Grant No. GG202404).

Conflict of interest

The authors declare that they have no conflict of interest.

References

1. F. Black, M. Scholes, The pricing of options and corporate liabilities, *J. Polit. Econ.*, **81** (1973), 637–654. http://doi.org/10.1086/260062
2. R. C. Merton, Theory of rational option pricing, *Bell J. Econ. Manag. Sci.*, **4** (1973), 141–183. http://doi.org/10.2307/3003143
3. S. L. Heston, A closed-form solution for options with stochastic volatility with applications to bond and currency options, *Rev. Financ. Stud.*, **6** (1993), 327–343. http://doi.org/10.1093/rfs/6.2.327
4. J. Gatheral, *The Volatility Surface: A Practitioner's Guide*, Hoboken: John Wiley & Sons, 2011. http://doi.org/10.1002/9781119202073
5. P. Carr, D. B. Madan, Option valuation using the fast Fourier transform, *J. Comput. Finance*, **2** (1999), 61–73. http://doi.org/10.21314/JCF.1999.043
6. F. Fang, C. W. Oosterlee, A novel pricing method for European options based on Fourier–cosine series expansions, *SIAM J. Sci. Comput.*, **31** (2009), 826–848. http://doi.org/10.1137/080718061
7. H. Albrecher, P. A. Mayer, W. Schoutens, J. Tistaert, The little Heston trap, *Wilmott*, **1** (2007), 83–92.
8. D. Tavella, C. Randall, *Pricing Financial Instruments: The Finite Difference Method*, New York: John Wiley & Sons, 2000.
9. D. J. Duffy, *Finite Difference Methods in Financial Engineering: A Partial Differential Equation Approach*, Chichester: John Wiley & Sons, 2006. http://doi.org/10.1002/9781118673447
10. B. Düring, J. Miles, High-order ADI scheme for option pricing in stochastic volatility models, *J. Comput. Appl. Math.*, **316** (2017), 109–121. http://doi.org/10.1016/j.cam.2016.09.040
11. P. Glasserman, *Monte Carlo Methods in Financial Engineering*, New York: Springer, 2004. http://doi.org/10.1007/978-0-387-21617-1
12. L. Andersen, Simple and efficient simulation of the Heston stochastic volatility model, *J. Comput. Finance*, **11** (2008), 1–42. http://doi.org/10.21314/JCF.2008.189
13. R. Lord, R. Koekkoek, D. van Dijk, A comparison of biased simulation schemes for stochastic volatility models, *Quant. Finance*, **10** (2010), 177–194. http://doi.org/10.1080/14697680802392496

14. Y. Cui, S. del Baño Rollin, G. Germano, Full and fast calibration of the Heston stochastic volatility model, *Eur. J. Oper. Res.*, **263** (2017), 625–638. http://doi.org/10.1016/j.ejor.2017.05.018
15. J. Sirignano, K. Spiliopoulos, DGM: A deep learning algorithm for solving partial differential equations, *J. Comput. Phys.*, **375** (2018), 1339–1364. http://doi.org/10.1016/j.jcp.2018.08.029
16. J. Han, A. Jentzen, W. E, Solving high-dimensional partial differential equations using deep learning, *Proc. Natl. Acad. Sci. USA*, **115** (2018), 8505–8510. http://doi.org/10.1073/pnas.1718942115
17. M. Raissi, P. Perdikaris, G. E. Karniadakis, Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, *J. Comput. Phys.*, **378** (2019), 686–707. http://doi.org/10.1016/j.jcp.2018.10.045
18. G. E. Karniadakis, I. G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, L. Yang, Physics-informed machine learning, *Nat. Rev. Phys.*, **3** (2021), 422–440. http://doi.org/10.1038/s42254-021-00314-5
19. A. D. Jagtap, G. E. Karniadakis, Extended physics-informed neural networks (XPINNs): A generalized space-time domain decomposition based deep learning framework for nonlinear partial differential equations, *Commun. Comput. Phys.*, **28** (2020), 2002–2041. http://doi.org/10.4208/cicp.OA-2020-0164
20. L. Lu, X. Meng, Z. Mao, G. E. Karniadakis, DeepXDE: A deep learning library for solving differential equations, *SIAM Rev.*, **63** (2021), 208–228. http://doi.org/10.1137/19M1274067
21. S. Cuomo, V. Schiano Di Cola, F. Giampaolo, G. Rozza, M. Raissi, F. Piccialli, Scientific machine learning through physics-informed neural networks: Where we are and what's next, *J. Sci. Comput.*, **92** (2022), 88. http://doi.org/10.1007/s10915-022-01939-z
22. A. Krishnapriyan, A. Gholami, S. Zhe, R. Kirby, M. W. Mahoney, Characterizing possible failure modes in physics-informed neural networks, In: *Advances in Neural Information Processing Systems*, **34** (2021), 26548–26560.
23. S. Wang, Y. Teng, P. Perdikaris, Understanding and mitigating gradient flow pathologies in physics-informed neural networks, *SIAM J. Sci. Comput.*, **43** (2021), A3055–A3081. http://doi.org/10.1137/20M1318043
24. S. Wang, X. Yu, P. Perdikaris, When and why PINNs fail to train: A neural tangent kernel perspective, *J. Comput. Phys.*, **449** (2022), 110768. http://doi.org/10.1016/j.jcp.2021.110768
25. L. D. McClenny, U. M. Braga-Neto, Self-adaptive physics-informed neural networks, *J. Comput. Phys.*, **474** (2023), 111722. http://doi.org/10.1016/j.jcp.2022.111722
26. C. Wu, M. Zhu, Q. Tan, Y. Kartha, L. Lu, A comprehensive study of non-adaptive and residual-based adaptive sampling for physics-informed neural networks, *Comput. Methods Appl. Mech. Eng.*, **403** (2023), 115671. http://doi.org/10.1016/j.cma.2022.115671
27. K. Tang, X. Wan, C. Yang, DAS-PINNs: A deep adaptive sampling method for solving high-dimensional partial differential equations, *J. Comput. Phys.*, **476** (2023), 111868. http://doi.org/10.1016/j.jcp.2022.111868
28. Z. Gao, L. Yan, T. Zhou, Failure-informed adaptive sampling for PINNs, *SIAM J. Sci. Comput.*, **45** (2023), A1971–A1994. http://doi.org/10.1137/22M1527763

29. Y. Bai, T. Chaolu, S. Bilige, The application of improved physics-informed neural network (IPINN) method in finance, *Nonlinear Dyn.*, **107** (2022), 3655–3667. http://doi.org/10.1007/s11071-021-07146-z
30. F. Gatta, V. Schiano Di Cola, F. Giampaolo, F. Piccialli, S. Cuomo, Meshless methods for American option pricing through physics-informed neural networks, *Eng. Anal. Bound. Elem.*, **151** (2023), 68–82. http://doi.org/10.1016/j.enganabound.2023.02.040
31. H. O. Bae, S. Kang, M. Lee, Option pricing and local volatility surface by physics-informed neural network, *Comput. Econ.*, **64** (2024), 3143–3171. http://doi.org/10.1007/s10614-024-10551-2
32. D. Hainaut, A. Casas, Option pricing in the Heston model with physics inspired neural networks, *Ann. Finance*, **20** (2024), 353–376. http://doi.org/10.1007/s10436-024-00452-7
33. E. Perez, F. Strub, H. de Vries, V. Dumoulin, A. Courville, FiLM: Visual reasoning with a general conditioning layer, In: *Proceedings of the AAAI Conference on Artificial Intelligence*, **32** (2018), 3942–3951. http://doi.org/10.1609/aaai.v32i1.11671
34. L. Lu, P. Jin, G. Pang, Z. Zhang, G. E. Karniadakis, Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators, *Nat. Mach. Intell.*, **3** (2021), 218–229. http://doi.org/10.1038/s42256-021-00302-5
35. Z. Li, N. Kovachki, K. Azizzadenesheli, B. Liu, K. Bhattacharya, A. Stuart, et al., Fourier neural operator for parametric partial differential equations, In: *Proceedings of the International Conference on Learning Representations*, 2021.
36. N. Kovachki, Z. Li, B. Liu, K. Azizzadenesheli, K. Bhattacharya, A. Stuart, et al., Neural operator: Learning maps between function spaces with applications to PDEs, *J. Mach. Learn. Res.*, **24** (2023), 1–97.
37. J. C. Cox, J. E. Ingersoll, S. A. Ross, A theory of the term structure of interest rates, *Econometrica*, **53** (1985), 385–407. http://doi.org/10.2307/1911242
38. K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, 770–778. http://doi.org/10.1109/CVPR.2016.90
39. N. Sukumar, A. Srivastava, Exact imposition of boundary conditions with distance functions in physics-informed deep neural networks, *Comput. Methods Appl. Mech. Eng.*, **389** (2022), 114333. http://doi.org/10.1016/j.cma.2021.114333
40. M. A. Nabian, R. J. Gladstone, H. Meidani, Efficient training of physics-informed neural networks via importance sampling, *Comput. Aided Civ. Infrastruct. Eng.*, **36** (2021), 962–977. http://doi.org/10.1111/mice.12685
41. J. Yu, L. Lu, X. Meng, G. E. Karniadakis, Gradient-enhanced physics-informed neural networks for forward and inverse PDE problems, *Comput. Methods Appl. Mech. Eng.*, **393** (2022), 114823. http://doi.org/10.1016/j.cma.2022.114823



AIMS Press

© 2026 the Author(s), licensee AIMS Press. This is an open access article distributed under the terms of the Creative Commons Attribution License (https://creativecommons.org/licenses/by/4.0)