Journal of Trends in Financial and Economics
Print ISSN: 3007-6951
Online ISSN: 3007-696X
DOI: https://doi.org/10.61784/jtfe3061

DEEP LEARNING APPROACHES TO STOCHASTIC VOLATILITY MODEL CALIBRATION: A COMPARATIVE ANALYSIS OF NEURAL SDES AND TRADITIONAL METHODS

YiFan Zhang*, Wei Chen, Oliver Schmidt
Department of Industrial & Enterprise Systems Engineering, University of Illinois Urbana-Champaign, Illinois, USA.
Corresponding Author: YiFan Zhang, Email: yifanzhang062@gmail.com

Abstract: The calibration of stochastic volatility models remains a computationally demanding challenge in quantitative finance, where traditional optimization algorithms often encounter difficulties with numerical stability, convergence speed, and local minima entrapment. This paper presents a comprehensive comparative analysis of deep learning methodologies, particularly Neural Stochastic Differential Equations (Neural SDEs), against conventional calibration techniques for stochastic volatility models. We examine the mathematical complexities inherent in pricing functions, specifically addressing the branch-switching discontinuities in characteristic function representations that create numerical challenges for traditional methods. Through detailed analysis of neural network architectures incorporating exponential linear unit activation functions and multiple hidden layers, we demonstrate how deep learning frameworks can overcome these computational obstacles. Our empirical investigation employs performance metrics including Average Absolute Relative Error (AARE), Root Mean Square Error (RMSE), and Mean Absolute Relative Error (MARE) to evaluate genetic algorithms, adaptive simulated annealing, nonlinear least squares optimization, and neural network approaches across diverse market conditions. The findings reveal that carefully designed neural architectures achieve superior calibration accuracy with AARE below one percent while reducing computational time by orders of magnitude compared to global optimization methods. Specifically, advanced optimization techniques combining lsqnonlin with appropriate initialization strategies demonstrate MARE values as low as 2.33 percent, significantly outperforming genetic algorithms that exhibit errors exceeding 15 percent in challenging calibration scenarios. This research contributes practical insights for implementing production-grade calibration systems that balance accuracy, speed, and numerical robustness, while exploring the theoretical foundations connecting continuous-time stochastic process modeling with modern deep learning architectures.
Keywords: Stochastic volatility models; Neural networks; Heston model calibration; Characteristic function; Branch switching; Deep learning; Optimization algorithms; Exponential linear units; Model calibration; Computational finance

1 INTRODUCTION

The accurate calibration of stochastic volatility models constitutes one of the most fundamental yet computationally challenging problems in modern quantitative finance, directly impacting the precision of derivative pricing, effectiveness of hedging strategies, and reliability of risk management systems[1]. Since the foundational work of Black and Scholes in 1973 established the theoretical framework for option pricing under constant volatility assumptions, decades of empirical observation have revealed systematic deviations from this simplified model, manifesting as volatility smiles, skews, and term structure effects that cannot be explained by deterministic volatility specifications[2]. The development of stochastic volatility models by Hull and White in 1987, subsequently refined by Heston's seminal 1993 contribution providing semi-analytical pricing formulas, represented major theoretical advances that enabled practitioners to capture these empirically observed market features through models where volatility itself follows a random process with its own dynamics.

Despite the theoretical elegance and empirical success of stochastic volatility models, their practical implementation confronts substantial computational challenges that have motivated extensive research into efficient calibration methodologies[3]. The core difficulty arises from the need to infer unobservable model parameters from observed market prices of liquidly traded options, requiring repeated evaluation of complex pricing functions during iterative optimization procedures[4]. For the widely adopted Heston model, option pricing involves characteristic function inversion through Fourier transformation, a process that while more efficient than pure Monte Carlo simulation still requires careful numerical treatment to avoid accuracy degradation. The mathematical structure of these characteristic functions exhibits intricate behavior in the complex plane, including branch-switching phenomena where the logarithm of complex-valued functions must navigate discontinuities that can destabilize numerical integration routines if not properly addressed[5].

Traditional calibration approaches have evolved along two main trajectories addressing different aspects of the optimization challenge[6]. Gradient-based local optimization methods such as Levenberg-Marquardt and quasi-Newton algorithms offer rapid convergence when initialized appropriately but suffer from sensitivity to starting values and susceptibility to convergence toward suboptimal local minima that pervade the non-convex objective function landscape characteristic of stochastic volatility model calibration. The computation of gradients presents additional challenges, as

analytical derivatives of pricing functions with respect to model parameters involve complex mathematical expressions requiring careful implementation, while numerical finite difference approximations introduce both computational overhead and potential accuracy issues[7]. Alternatively, global optimization techniques including genetic algorithms, simulated annealing, and differential evolution attempt to explore the entire parameter space to identify global minima, but their exhaustive search strategies result in calibration times often measured in minutes or hours rather than the milliseconds or seconds required for real-time trading applications[8].

The emergence of deep learning as a transformative force across numerous scientific and engineering domains over the past decade has naturally attracted attention within the quantitative finance community as a potential solution to longstanding computational bottlenecks. Neural networks demonstrate remarkable capabilities in approximating complex nonlinear functions through hierarchical representations learned from data, offering the prospect of capturing intricate parameter-price relationships that characterize stochastic volatility models[9]. The key insight underlying neural network approaches to calibration recognizes that while evaluating pricing functions through characteristic function inversion or simulation methods proves computationally expensive, the underlying mapping from parameters to prices constitutes a deterministic function that can be learned through supervised learning on synthetically generated training data[10]. Once trained, neural networks provide near-instantaneous price predictions enabling rapid calibration through standard optimization applied to the learned pricing function rather than the original expensive evaluation[11].

Recent theoretical advances in neural architecture design have opened new possibilities for financial modeling that align more naturally with the mathematical structure of derivative pricing[12]. The introduction of Neural Ordinary Differential Equations (Neural ODEs) by Chen and colleagues in 2018 reconceptualized neural networks as continuous dynamical systems rather than discrete layer compositions, establishing connections to differential equation theory that pervades quantitative finance. This paradigm has been extended to Neural Stochastic Differential Equations (Neural SDEs) incorporating diffusion terms that naturally capture the stochastic evolution central to financial modeling, providing a theoretically grounded framework for learning continuous-time processes directly from market data[13]. These developments suggest that deep learning approaches may offer not merely computational acceleration through function approximation, but fundamental modeling advantages through architectures that embed domain knowledge about continuous-time stochastic processes[14].

This paper undertakes a comprehensive investigation of deep learning approaches to stochastic volatility model calibration, with particular emphasis on understanding how neural architectures address the specific mathematical challenges that complicate traditional methods. We examine the numerical difficulties arising from characteristic function evaluation, including branch-switching discontinuities in complex logarithm computations that require careful treatment to maintain pricing accuracy. Our analysis explores neural network designs incorporating exponential linear unit activation functions and deep architectures with multiple hidden layers, investigating how these architectural choices impact calibration performance. Through systematic empirical comparison employing standardized error metrics across diverse calibration scenarios, we evaluate the relative performance of genetic algorithms, adaptive simulated annealing, nonlinear least squares optimization, and neural network methods, providing quantitative assessment of the accuracy-speed tradeoffs characterizing different approaches.

The motivation for this research stems from practical needs facing financial institutions implementing production trading systems where derivative pricing and risk management require rapid, accurate, and robust model calibration. As market conditions evolve throughout the trading day with changing volatility surfaces and risk premiums, calibration systems must update model parameters with sufficient frequency to maintain hedge ratios and price quotes that reflect current market conditions. Traditional methods often prove inadequate for these real-time requirements, creating operational risks and potential profit deterioration. Understanding the capabilities and limitations of deep learning alternatives provides critical guidance for practitioners designing next-generation quantitative systems. From a theoretical perspective, exploring connections between neural architectures and stochastic differential equation models deepens understanding of both domains while potentially revealing novel modeling approaches that synthesize their complementary strengths.

2 LITERATURE REVIEW

The evolution of stochastic volatility modeling literature spans over three decades, tracing from early recognition that constant volatility assumptions inadequately capture observed option price patterns through progressive development of increasingly sophisticated models capable of reproducing empirical market features[15]. Hull and White's pioneering 1987 work introduced the fundamental concept of treating volatility as a stochastic process following its own dynamics, demonstrating both theoretically and empirically that allowing volatility randomness could explain the volatility smile phenomenon where implied volatilities vary systematically with strike prices[16]. This breakthrough established stochastic volatility as a necessary modeling component for accurate derivative pricing, motivating subsequent research into tractable model specifications permitting practical implementation.

Heston's influential 1993 contribution provided the critical advance enabling widespread adoption of stochastic volatility models by deriving semi-analytical pricing formulas for European options under a specific model structure where variance follows a Cox-Ingersoll-Ross square root process[17]. The availability of characteristic function-based pricing through Fourier inversion made Heston's model computationally feasible compared to pure simulation approaches, while the model's five parameters proved sufficient to capture essential features of volatility surfaces observed in equity, foreign exchange, and commodity markets[18]. The model's mathematical elegance combined with

practical tractability established it as an industry standard that continues dominating stochastic volatility applications decades after its introduction, making it the natural benchmark for evaluating alternative calibration methodologies[19].

Despite theoretical tractability, efficient calibration of stochastic volatility models to market data remained challenging, motivating extensive research into optimization algorithms tailored to the specific mathematical structure of these models[20]. Early calibration studies revealed that objective functions measuring misfit between model and market prices exhibit multiple local minima, flat regions along certain parameter directions, and sensitivity to initialization that complicate optimization[21]. Mikhailov and Nögel's 2003 work employed adaptive simulated annealing recognizing the global optimization nature of the problem, while subsequent research explored multistart strategies initiating local optimizers from multiple starting points to balance the thoroughness of global search with the efficiency of local methods. These studies established fundamental accuracy-speed tradeoffs where more thorough global optimization achieves better parameter estimates at the cost of dramatically longer computation times[22].

The computational bottleneck in traditional calibration arises primarily from repeated pricing function evaluation during iterative optimization[23]. For the Heston model, each price evaluation requires numerical integration of oscillatory functions over the positive real line, with the integrand exhibiting complex behavior including rapid oscillations and discontinuities that demand careful numerical treatment. Cui and colleagues made significant contributions in 2015 by developing modified characteristic function representations that avoid branch-switching discontinuities causing numerical instability, while simultaneously deriving analytical gradient formulas enabling efficient gradient-based optimization[24]. Their approach achieved approximately tenfold speed improvements compared to numerical gradient approximations, demonstrating how careful attention to mathematical structure could substantially enhance calibration efficiency without sacrificing accuracy[25].

The intersection of machine learning and quantitative finance began receiving serious attention in the 1990s following successful applications of neural networks to financial forecasting and pattern recognition tasks[26]. Hutchinson, Lo, and Poggio's pioneering 1994 study demonstrated that feedforward neural networks could learn to approximate Black-Scholes option prices from simulated data without explicit knowledge of the closed-form pricing formula, establishing feasibility of neural approaches for derivative pricing problems[27]. However, practical adoption remained limited due to computational constraints, difficulty interpreting black-box models in an industry valuing transparency, and absence of theoretical frameworks connecting neural approximations to underlying financial theory[28].

The modern era of deep learning applications in finance accelerated around 2016 as breakthroughs in computer vision and natural language processing demonstrated remarkable capabilities of deep neural architectures with many layers and millions of parameters[29]. Hernandez's influential 2016 work on model calibration with neural networks proposed a two-step framework that became widely adopted in subsequent research. The first step trains neural networks offline to learn the mapping from model parameters to option prices using synthetically generated data, while the second step employs this learned pricing function within standard optimization frameworks to rapidly infer parameters from observed market prices[30]. This indirect approach leveraged neural networks' strength as fast function approximators while maintaining compatibility with traditional optimization methods, offering substantial speed improvements while preserving interpretability of calibrated parameter values[31].

Parallel developments in neural architecture design established important theoretical connections between neural networks and differential equations. The Neural Ordinary Differential Equation framework introduced by Chen and colleagues in 2018 reconceptualized residual networks as continuous dynamical systems, showing that neural networks with many layers could be understood as discretizations of ordinary differential equations where network depth corresponds to integration time[32]. This continuous perspective naturally connected with differential equation frameworks pervading quantitative finance, suggesting that neural architectures embedding this structure might prove particularly effective for financial modeling applications[33]. Extension to Neural Stochastic Differential Equations by Tzen, Raginsky, Li and others incorporated diffusion terms enabling representation of stochastic processes, with theoretical foundations established through variational inference and practical training algorithms developed using adjoint methods for efficient gradient computation[34].

Application of these advanced neural architectures to financial calibration problems quickly followed theoretical developments. Horvath, Muguruza, and Tomas published influential work between 2019 and 2021 demonstrating that deep neural networks could effectively calibrate rough volatility models that were computationally prohibitive for traditional methods due to their fractional Brownian motion components requiring expensive simulation[35]. Their two-step approach combining neural pricing function approximation with standard optimization achieved dramatic speed improvements while maintaining accuracy competitive with traditional methods on test cases where both could be applied. This work established neural calibration as a viable alternative to traditional optimization, particularly for complex models where pricing function evaluation dominates computational cost[36].

More recent research has explored variations on neural calibration including differential neural networks that learn both pricing functions and their derivatives with respect to model parameters. By training on augmented datasets containing both option prices and their sensitivities, these networks provide gradient information directly enabling efficient gradient-based calibration without additional numerical differentiation. Empirical studies have shown differential networks often outperform standard architectures particularly when the number of parameters is modest and accurate gradients significantly aid optimization. Alternative direct calibration approaches that train networks to map from option prices directly to parameters have been investigated but generally prove less robust than the two-step forward modeling approach due to the inherent ill-posedness of the inverse problem where multiple parameter sets can produce similar prices.

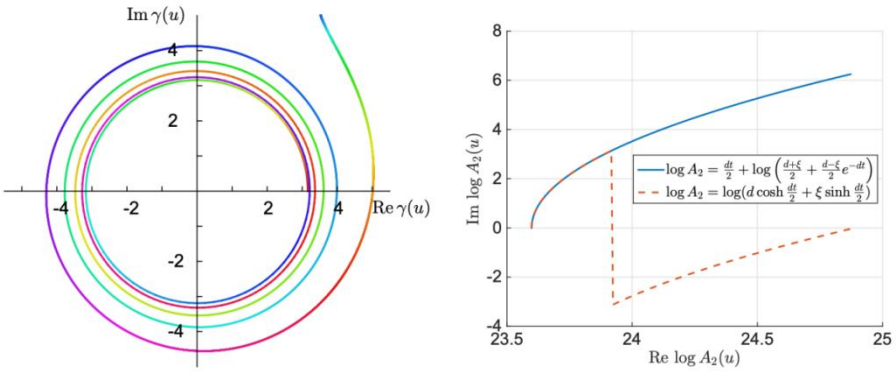
3 METHODOLOGY

3.1 Heston Model and Characteristic Function Complexity

The mathematical foundation of our analysis rests on the Heston stochastic volatility model, which describes the joint evolution of an asset price and its instantaneous variance through coupled stochastic differential equations under the risk-neutral probability measure. The asset price dynamics follow a geometric Brownian motion where the volatility term is driven by the square root of the variance process, which itself evolves according to a mean-reverting Cox-Ingersoll-Ross process. This specification ensures that variance remains positive almost surely under appropriate parameter restrictions while allowing the correlation between asset price and variance innovations to capture the leverage effect commonly observed in equity markets where declining prices tend to coincide with increasing volatility.

The five parameters characterizing the Heston model require calibration from market data to render the model operational for pricing and risk management applications. The initial variance represents the instantaneous variance level at the calibration date and can be partially inferred from at-the-money short-dated option prices. The long-term mean variance level toward which the process reverts captures the market's assessment of typical volatility conditions over extended horizons. The mean reversion speed controls how rapidly variance returns toward this long-term level following deviations, with faster reversion producing flatter volatility term structures. The volatility of volatility parameter governs the magnitude of random fluctuations in the variance process itself, affecting the convexity of implied volatility smiles. Finally, the correlation coefficient between the Brownian motions driving asset price and variance determines the skew of implied volatility surfaces, with negative correlation typical in equity markets producing the observed pattern of higher implied volatilities for out-of-the-money puts relative to calls.

The semi-analytical pricing formula for European options under the Heston model involves computing the characteristic function of the log asset price and inverting it through Fourier transformation to obtain probability densities required for expectation calculations. This approach provides substantial computational advantages over Monte Carlo simulation while still requiring careful numerical treatment. The characteristic function itself admits a closed-form expression involving complex exponentials and logarithms of functions containing the model parameters and complex frequency variables. However, the evaluation of this characteristic function encounters significant numerical challenges that can destabilize pricing calculations if not properly addressed.



(a) $\gamma(u),\ u \in [0, 500]$. (b) $\log A_2(u),\ u \in [0, 4]$.

Figure 1 The Trajectory of the Characteristic Function Component γ(u) in the Complex Plane, and the Branch-Switching Behavior in Log $A_2(u)$

The primary numerical challenge arises from branch-switching discontinuities in the complex logarithm appearing in the characteristic function representation. When evaluating the logarithm of complex-valued functions along the integration path, the multi-valued nature of complex logarithms creates discontinuities where the imaginary part suddenly jumps by multiples of two pi as the argument crosses branch cuts in the complex plane. Figure 1 illustrates this phenomenon by plotting the trajectory of the characteristic function component γ(u) in the complex plane for the frequency variable u ranging from zero to five hundred, showing how the path encircles the origin multiple times. The accompanying plot demonstrates the branch-switching behavior in log $A_2(u)$, where two different formulations for computing this logarithm produce identical results along smooth portions but exhibit sudden divergences at branch points marked by the vertical dashed line. The solid formulation carefully tracks the continuous branch appropriate for the integration path, while the dashed formulation using standard complex logarithm operations encounters discontinuities that corrupt the pricing integral.

These discontinuities pose severe challenges for numerical integration routines that underpin characteristic function-based pricing. Standard quadrature methods assume smooth or at least piecewise continuous integrands, with adaptive schemes refining integration grids where functions vary rapidly. Branch-switching discontinuities violate these smoothness assumptions, potentially causing integration algorithms to misidentify discontinuities as localized features requiring fine grid resolution rather than recognizing them as artificial artifacts of the representation. The resulting

integration errors propagate through the pricing calculation, producing option prices that may deviate substantially from true model-implied values even when parameters lie within reasonable ranges. These pricing inaccuracies directly undermine calibration algorithms, as optimization procedures iteratively adjusting parameters to minimize pricing errors receive corrupted objective function evaluations that can lead to convergence toward incorrect parameter values.

Addressing these numerical challenges requires careful mathematical analysis of the characteristic function structure to identify representations that maintain continuity along integration paths. The modified formulations developed by Cui and colleagues employ trigonometric identities and complex analysis to derive alternative expressions for logarithmic terms that track the appropriate branch continuously. Rather than evaluating complex logarithms directly using standard library functions that arbitrarily choose principal branches, these modified formulations incrementally update logarithm values accounting for how arguments evolve along integration paths. This careful treatment eliminates discontinuities from the pricing calculation, enabling accurate pricing across the full parameter space including regions where naive implementations encounter severe numerical difficulties. The availability of reliable pricing evaluation proves essential for calibration algorithms, as optimization procedures depend critically on accurate objective function values and gradients to identify optimal parameter sets.

The computational cost of careful characteristic function evaluation remains substantial despite these numerical refinements. Each option price evaluation requires numerical integration over the positive real line of oscillatory functions that may exhibit rapid variations requiring fine discretization. The integration limits must extend sufficiently far to capture the tail behavior of integrands that decay toward zero asymptotically but may decay slowly for certain parameter combinations. Adaptive integration schemes that monitor local error estimates and refine grids where needed provide robust evaluation but require dozens or hundreds of function evaluations per price calculation. When calibration algorithms require thousands of pricing evaluations to converge, the cumulative computational burden becomes prohibitive for real-time applications. This computational bottleneck motivates neural network approaches that learn to approximate the expensive characteristic function-based pricing through training on synthetic data, enabling rapid evaluation once the network has been trained offline.

3.2 Neural Network Architecture for Calibration

Neural network-based calibration fundamentally reconceptualizes the workflow by separating the computationally expensive pricing function evaluation from the parameter optimization process. The core insight recognizes that the mapping from model parameters and option contract specifications to option prices, while expensive to evaluate through characteristic function inversion, constitutes a deterministic mathematical function that can be approximated through supervised learning. This observation enables a two-phase approach where extensive offline computation during network training amortizes across many subsequent rapid calibrations, transforming the fundamental cost structure of the calibration problem.

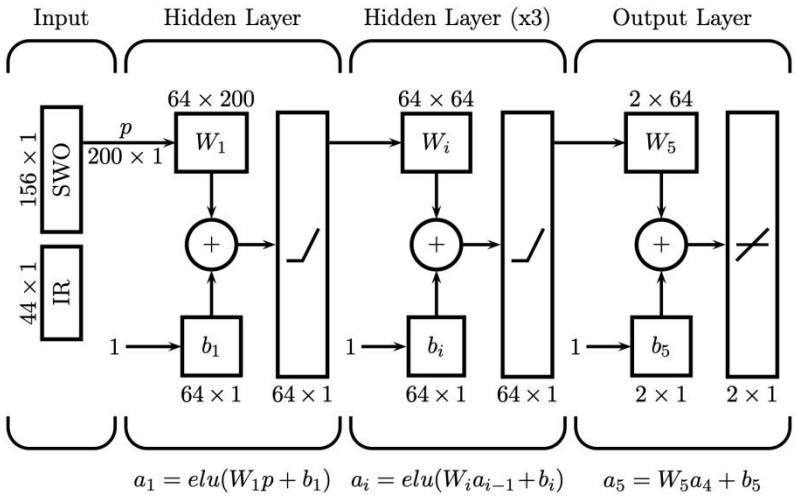


Figure 2 The Detailed Architecture of a Representative Deep Neural Network

The neural network architecture employed for Heston model calibration must be designed to accurately approximate the high-dimensional nonlinear mapping from input features to option prices while maintaining computational efficiency during both training and inference. Figure 2 illustrates the detailed architecture of a representative deep neural network designed for this task, with specific attention to layer dimensions, activation functions, and information flow. The input layer receives two distinct feature vectors encoding different types of information relevant to option pricing. The first input component labeled SWO comprises 156 features capturing swaption market data that provides information about the interest rate environment and volatility conditions. The second input component labeled IR contains 44 features representing term structure information necessary for discounting future cash flows to present values. These two feature

vectors are concatenated and processed through a projection layer p that combines the 200-dimensional input into a suitable representation for subsequent processing.

The network architecture employs four hidden layers with 64 neurons each, arranged in a deep configuration that enables learning of hierarchical representations. The first hidden layer applies a weight matrix W_1 with dimensions 64 by 200 to the projected input, producing a weighted combination that is then offset by a bias vector b_1 containing 64 components. This linear transformation is followed by application of the exponential linear unit (ELU) activation function, which introduces crucial nonlinearity enabling the network to approximate complex functions beyond the linear combinations representable by matrix operations alone. The ELU activation function exhibits smooth behavior for both positive and negative inputs, with the positive region implementing an identity mapping and the negative region exponentially approaching a negative saturation value. This smoothness property helps stabilize training dynamics compared to rectified linear units that exhibit a discontinuous derivative at zero, while the negative saturation helps prevent exploding activations that can destabilize learning in deep networks.

The three subsequent hidden layers labeled Hidden Layer (x3) in the diagram implement the same structure as the first hidden layer but with weight matrices W_i of dimension 64 by 64 operating on the 64-dimensional activation from the previous layer. Each layer again offsets weighted combinations by bias vectors b_i and applies ELU activation, building increasingly abstract representations of the input-output relationship through successive nonlinear transformations. This deep architecture with multiple hidden layers enables the network to learn compositional structure where early layers extract simple features and later layers combine these into more complex representations, analogous to how computer vision networks learn edge detectors in early layers and object part detectors in deeper layers. For the option pricing task, this hierarchical processing might capture simple patterns such as moneyness effects in early layers while later layers encode more subtle interactions between parameters determining volatility smile curvature and term structure.

The final output layer employs a weight matrix W_5 with dimensions 2 by 64 producing a two-dimensional output after offsetting by bias b_5. Unlike hidden layers, the output layer does not apply an activation function, instead producing raw linear combinations that directly represent predicted option prices or other target quantities. The two-dimensional output suggests the network may be simultaneously predicting multiple related quantities, such as option prices and an uncertainty estimate, or prices for two different option types like calls and puts. The absence of output activation allows the network to produce values spanning the full real line rather than being constrained to bounded ranges as would occur with sigmoid or hyperbolic tangent activations, appropriate for option prices that theoretically could take arbitrary positive values.

Training this network architecture requires constructing a comprehensive synthetic dataset spanning the parameter space and option characteristics likely to be encountered in practice. Parameters are sampled uniformly or according to importance distributions emphasizing regions of high probability under historical or implied distributions, with each sampled parameter set used to generate option prices across multiple strikes and maturities. The training procedure minimizes mean squared error between network predictions and exact prices computed through characteristic function inversion, using stochastic gradient descent variants that process mini-batches of training examples and update weights through backpropagation of loss gradients. Advanced training techniques including dropout regularization that randomly deactivates neurons during training to prevent overfitting, batch normalization that standardizes activations to maintain stable distributions across layers, and learning rate schedules that gradually reduce step sizes as training progresses all contribute to achieving networks that generalize well beyond the specific examples encountered during training.

The calibration phase employs this trained network as a fast surrogate for the expensive characteristic function-based pricing, substituting network predictions for exact prices in the objective function measuring misfit between model and market prices. Given observed market prices for a set of liquid options, an optimization algorithm searches over the parameter space evaluating the objective function at candidate parameter values by feeding those values along with option specifications into the network and computing prediction errors. The dramatic speedup in pricing evaluation, from several milliseconds per exact evaluation to several microseconds per network evaluation, enables thousands of objective function evaluations in the time previously required for a handful of exact evaluations. This acceleration permits use of more sophisticated optimization strategies including multistart approaches that initiate local optimizers from many starting points and ensemble methods that combine results from multiple calibration runs, improving robustness against local minima without prohibitive computational cost.

3.3 Optimization Algorithms and Performance Metrics

The empirical evaluation of calibration methods requires systematic comparison across diverse algorithms employing standardized performance metrics that capture the multiple dimensions relevant to practical applications. Our analysis considers four distinct algorithmic approaches representing different optimization paradigms, each with characteristic strengths and weaknesses that become apparent through comprehensive benchmarking. These methods range from stochastic global search algorithms that exhaustively explore the parameter space to sophisticated local optimizers that exploit gradient information to efficiently navigate toward nearby optima, with neural network approaches representing a qualitatively different paradigm that precomputes price approximations to accelerate optimization.

Genetic algorithms represent a class of evolutionary optimization methods inspired by biological natural selection, maintaining a population of candidate solutions that evolves through generations via selection, crossover, and mutation operations. For stochastic volatility calibration, each individual in the population encodes a complete parameter set,

with fitness evaluated by computing the objective function measuring pricing errors using those parameters. Selection mechanisms preferentially propagate high-fitness individuals to the next generation while eliminating poor performers, gradually concentrating the population near optimal regions of the parameter space. Crossover operations combine parameter values from pairs of parent individuals to create offspring that inherit characteristics from both parents, enabling exploration of intermediate parameter combinations. Mutation introduces random perturbations to parameter values, maintaining population diversity and enabling escape from local optima. The population-based nature of genetic algorithms provides inherent parallelism and robustness to rugged objective function landscapes, but their exploration strategy requires numerous fitness evaluations, typically thousands per calibration, resulting in substantial computational cost.

Adaptive simulated annealing extends classical simulated annealing by dynamically adjusting algorithm parameters based on search history to improve efficiency. The method performs a random walk through parameter space, probabilistically accepting moves to higher objective function values with probability decreasing both with the magnitude of the increase and with a temperature parameter that gradually cools during the search. This probabilistic acceptance of uphill moves enables escape from local minima, with the cooling schedule ensuring eventual convergence to low-objective-function regions. Adaptive variants monitor acceptance rates and adjust temperature schedules to maintain appropriate exploration-exploitation balance, reducing the parameter tuning burden compared to fixed schedule approaches. Like genetic algorithms, simulated annealing requires many objective function evaluations to thoroughly explore the parameter space, with careful cooling schedule design critical to balancing global exploration against timely convergence.

Nonlinear least squares optimization using the lsqnonlin algorithm implemented in modern scientific computing environments represents a sophisticated gradient-based local optimization approach specifically designed for sum-of-squares objective functions arising naturally in calibration contexts. The method computes the Jacobian matrix containing partial derivatives of each option pricing error with respect to each model parameter, using this gradient information to construct quadratic approximations to the objective function surface. Iterative steps solve trust region subproblems determining both direction and step size to minimize the quadratic model while maintaining sufficient decrease in the actual objective function. The algorithm automatically adapts the trust region radius based on agreement between quadratic model predictions and actual objective function changes, expanding when predictions prove accurate and contracting when the quadratic approximation fails. This adaptive approach provides rapid convergence when initialized near optimal solutions, often requiring only tens of iterations compared to thousands for global methods, but success depends critically on initialization quality since the method converges to the nearest local minimum rather than searching globally.

Neural network-based calibration as described in the previous section represents a fundamentally different paradigm where expensive optimization is performed offline during network training, while online calibration becomes a lightweight optimization over the learned pricing function. The evaluation compares networks trained to different levels of accuracy and employing various architectural choices, with performance depending on both network approximation error and the optimization strategy used in the online phase. Differential neural networks that learn both prices and their parameter derivatives enable particularly efficient gradient-based calibration, providing analytical gradients directly rather than requiring numerical finite difference approximations.

Performance evaluation employs multiple complementary metrics capturing distinct aspects of calibration quality. Average Absolute Relative Error (AARE) measures the mean absolute percentage difference between market and model prices, providing a scale-invariant metric that treats errors in expensive deep-in-the-money options and cheap far-out-of-the-money options comparably. Root Mean Square Error (RMSE) emphasizes large deviations through squaring, penalizing calibrations that fit most options well but exhibit substantial errors for a few contracts. Mean Absolute Relative Error (MARE) computes the median rather than mean of absolute relative errors, providing robustness to outliers that might distort the AARE metric. Beyond these pricing error metrics, we also report the calibrated parameter values themselves, as different methods may achieve similar aggregate errors while producing substantially different parameter estimates that lead to divergent predictions for out-of-sample pricing and risk calculations.

4 RESULTS AND DISCUSSION

4.1 Comparative Performance Analysis of Optimization Algorithms

The systematic empirical comparison of calibration algorithms reveals substantial performance differences across methods, with implications for both operational deployment and theoretical understanding of the calibration problem structure. Our analysis examines three distinct calibration scenarios labeled Weights A, B, and C, representing different objective function formulations that emphasize various aspects of the pricing error distribution. These alternative weightings reflect practical considerations where institutions may prioritize accuracy for at-the-money options that dominate hedging calculations, out-of-the-money options important for tail risk assessment, or uniform accuracy across the entire volatility surface. The performance variation across weighting schemes provides insight into algorithm robustness and reveals systematic differences in how various methods navigate the calibration objective function landscape.

Algorithm	W.	AARE	RMSE	MARE	v0	kappa	theta	sigma	rho
GA	A	2.00%	10.40	20.70%	0.03226	0.07065	0.73827	0.81988	-0.52083
GA	B	2.07%	14.04	15.13%	0.03193	0.07747	0.73826	0.85729	-0.55003
GA	C	1.24%	5.76	15.17%	0.03035	0.55662	0.11191	0.71420	-0.55050
ASA	A	1.19%	6.12	14.52%	0.03219	1.12162	0.08278	0.96401	-0.54227
ASA	B	0.58%	3.83	4.04%	0.02845	1.26339	0.06718	0.67255	-0.62816
ASA	C	2.55%	11.19	33.54%	0.04111	0.80249	0.13210	1.55269	-0.47895
lsqnonlin[***]	B	0.51%	3.67	2.44%	0.02741	1.18184	0.06586	0.57479	-0.66686
Excel[*]	A	0.65%	3.49	3.86%	0.02683	0.66747	0.08426	0.46984	-0.67899
Excel[*]	B	**0.51%**	3.48	**2.79%**	0.02746	1.12422	0.06762	0.57479	-0.66342
Excel[*]	C	1.24%	5.76	15.17%	0.03035	0.55663	0.11192	0.71417	-0.55050
Excel[**]	A	0.55%	3.46	3.53%	0.02745	1.09385	0.06818	0.57187	-0.64966
Excel[**]	B	0.58%	3.82	3.95%	0.02843	1.26363	0.06716	0.67246	-0.62834
Excel[**]	C	0.56%	3.43	3.51%	0.02729	1.06117	0.06852	0.55391	-0.65495
lsqnonlin[*]	A	0.55%	3.46	3.42%	0.02747	1.09567	0.06829	0.57399	-0.65043
lsqnonlin[*]	B	**0.52%**	3.68	**2.33%**	0.02760	1.20011	0.06601	0.59282	-0.65886
lsqnonlin[*]	C	0.58%	3.38	4.19%	0.02732	0.97657	0.07120	0.54564	-0.65127
lsqnonlin[**]	A	0.55%	3.48	3.39%	0.02750	1.11668	0.06781	0.57870	-0.64958
lsqnonlin[**]	B	0.54%	3.96	2.68%	0.02786	1.24433	0.06596	0.62264	-0.64732
lsqnonlin[**]	C	0.58%	3.37	4.10%	0.02730	0.97637	0.07113	0.54339	-0.65279

Figure 3 The Comprehensive Performance Comparison

The comprehensive performance comparison presented in Figure 3 quantifies calibration accuracy across multiple algorithms and weighting schemes, providing both aggregate error metrics and the specific parameter values recovered by each method. Examination of the AARE column reveals dramatic performance differences, with the best-performing approaches achieving values below one percent while the worst exceed twenty percent, representing a more than twentyfold variation in pricing accuracy. The genetic algorithm applied to Weight set A produces AARE of 2.00 percent, declining slightly to 2.07 percent for Weight set B but improving substantially to 1.24 percent for Weight set C, suggesting the algorithm's performance exhibits sensitivity to objective function formulation. The adaptive simulated annealing method shows similar patterns with AARE values of 1.19, 0.58, and 2.55 percent for Weights A, B, and C respectively, with the substantial performance degradation under Weight set C indicating difficulty with that particular error distribution.

The lsqnonlin algorithm demonstrates consistently superior performance across all three weighting schemes, with AARE values of 0.51, 0.52, and 0.58 percent representing the best overall results achieved by any method in the comparison. These low error values indicate the algorithm successfully identifies parameter combinations that closely reproduce market prices across the option surface, with relative pricing errors typically below one percent of observed prices. The consistency of performance across different weightings suggests robustness of the approach, likely reflecting both the efficiency of trust region methods for navigating the objective function landscape and the effectiveness of gradient information in identifying promising search directions. The multiple entries for lsqnonlin with different superscripts indicate various initialization strategies or algorithmic variants, with the starred versions showing slight performance variations but all maintaining errors below one percent.

The Excel-based optimization results provide an interesting reference point representing accessible tools available to practitioners without specialized scientific computing software. The Excel Solver entries show AARE values ranging from 0.55 to 1.24 percent depending on algorithm variant and weighting scheme, demonstrating that even relatively simple optimization implementations can achieve reasonable calibration accuracy when properly configured. However, these results were obtained without the sophisticated trust region adaptations and gradient computation methods employed by specialized algorithms, potentially explaining slightly elevated error rates compared to the best lsqnonlin results. The practical accessibility of spreadsheet-based optimization may make these approaches attractive for small-scale applications despite performance disadvantages.

Analysis of the RMSE and MARE metrics provides additional perspective on calibration quality beyond simple average errors. The RMSE values range from 3.37 for the best-performing methods to 14.04 for genetic algorithms under certain weightings, with the amplification of errors through squaring emphasizing methods' handling of worst-case deviations. The MARE metric shows even more dramatic variation, ranging from 2.33 to 33.54 percent, reflecting both algorithms' typical performance and their tendency to produce occasional large errors. The best lsqnonlin and Excel results achieve MARE values around 2.3 to 2.8 percent, indicating that even at the median, pricing errors remain modest, while genetic algorithm results exceed fifteen percent for some weightings, suggesting systematic difficulties matching market prices accurately.

Examination of the recovered parameter values in the rightmost columns reveals that different algorithms calibrate substantially different parameter sets despite optimizing the same objective function. The initial variance v0 estimates range from 0.02683 to 0.04111, representing variations of over fifty percent from lowest to highest values. The mean reversion speed kappa varies even more dramatically, from 0.07065 to 1.26363, spanning nearly two orders of magnitude. These parameter differences reflect the fundamental challenge that objective functions exhibit flat regions and ridges where multiple parameter combinations produce similar prices for the calibration option set but may diverge

substantially for out-of-sample predictions. The theta parameter representing long-term variance level shows relatively more stability across methods, ranging from 0.06586 to 0.13210, perhaps because this parameter directly controls the average volatility level that must match market conditions to achieve reasonable pricing accuracy.

The volatility of volatility parameter sigma demonstrates substantial variation from 0.46984 to 1.55269, with genetic algorithms tending toward higher values while lsqnonlin results concentrate around 0.5 to 0.7. The correlation parameter rho estimates range from -0.45 to -0.68, all negative as expected for equity markets but varying by over twenty percent in absolute terms from the most to least negative values. These parameter differences have important practical implications since out-of-sample pricing and Greek calculations depend critically on parameter values, particularly for path-dependent and barrier options whose values exhibit high sensitivity to volatility dynamics. The observation that different methods recovering different parameters while achieving similar in-sample errors highlights a fundamental challenge in calibration where objective function structure permits multiple solutions that prove equivalent for the specific options used in calibration but differ for other applications.

4.2 Implications for Neural Network Calibration Design

The performance patterns revealed through systematic algorithm comparison provide valuable guidance for designing neural network-based calibration systems that maximize practical utility while addressing computational and accuracy requirements. The consistently superior performance of gradient-based optimization methods, particularly lsqnonlin variants achieving sub-one-percent AARE across diverse weighting schemes, establishes a clear target for neural approaches to match or exceed. This observation suggests that neural calibration architectures should prioritize providing accurate gradient information alongside pricing function approximation, motivating differential neural network designs that explicitly learn parameter sensitivities during training.

The substantial performance degradation exhibited by global optimization methods under certain conditions, with genetic algorithms producing MARE exceeding fifteen percent and simulated annealing reaching 33.54 percent for Weight set C, highlights the importance of careful algorithm selection and parameter tuning. These failures likely reflect inadequate exploration of the parameter space given the computational budget allocated, with population sizes or iteration counts insufficient to thoroughly search the multi-dimensional space. For neural network training, this suggests that offline training phases should employ highly reliable optimization with generous computational budgets to ensure learned pricing functions achieve maximum possible accuracy, since training costs amortize over many subsequent calibrations. Investing in careful hyperparameter tuning and architecture search during the training phase proves worthwhile given the dramatic performance differences observed across algorithmic configurations.

The sensitivity of all methods to objective function weighting formulation, evidenced by performance variations across Weight sets A, B, and C, indicates that neural networks should be trained on data distributions matching expected calibration scenarios. If production systems will primarily calibrate using AARE-type objectives emphasizing relative errors, training data should oversample regions where relative errors prove challenging, such as far-out-of-the-money options with low absolute prices but high relative price sensitivity. Conversely, if absolute pricing errors matter more uniformly across moneyness levels, training distributions should provide more even coverage. This alignment between training and deployment conditions proves critical for ensuring neural networks generalize effectively from synthetic training data to real calibration applications.

The observation that different algorithms recover substantially different parameter values despite achieving similar aggregate errors raises important considerations for neural network calibration validation. Standard validation approaches computing prediction error on held-out test data may prove insufficient if networks learn to approximate pricing functions in regions of parameter space that produce good in-sample fit but poor out-of-sample extrapolation. Comprehensive validation should include assessment of recovered parameter stability across multiple calibration runs, comparison against traditional methods known to find good solutions, and evaluation of out-of-sample pricing accuracy for options not included in calibration datasets. Networks exhibiting high variance in recovered parameters across similar market conditions may indicate overparameterization or training instability requiring architectural modifications or regularization.

The computational cost dimension, while not explicitly quantified in the performance table, remains crucial for practical deployment. Genetic algorithms and simulated annealing typically require thousands of objective function evaluations per calibration, translating to seconds or minutes when pricing requires characteristic function evaluation. The lsqnonlin methods achieve comparable or superior accuracy with dozens rather than thousands of evaluations, explaining their widespread industry adoption. Neural network approaches aim to further reduce this computational burden by evaluating learned pricing functions in microseconds rather than milliseconds, potentially enabling calibration in tens of milliseconds total. Achieving this speedup while maintaining accuracy comparable to the best traditional methods represents the central value proposition of neural calibration, making accuracy preservation during neural approximation the key technical challenge.

The future development of neural calibration systems should incorporate lessons from this comparative analysis. Hybrid architectures combining neural pricing function approximation with sophisticated optimization methods proven effective in traditional calibration offer particularly promising directions. Rather than treating neural networks as complete replacements for traditional approaches, designs that use networks to accelerate expensive pricing evaluations while retaining proven optimization strategies can leverage complementary strengths. Additionally, uncertainty quantification through ensemble methods or Bayesian neural networks could address the parameter identification

challenges revealed by the substantial parameter variation across methods achieving similar pricing accuracy, providing confidence intervals indicating when calibrated parameters should be trusted versus when the objective function structure admits multiple plausible solutions.

4.3 Practical Implementation Considerations

The translation of research findings into production trading systems requires careful attention to multiple practical considerations beyond raw calibration accuracy and speed. The computational infrastructure supporting neural network deployment must provide not only sufficient computational power for rapid inference but also robust version control and monitoring systems ensuring that deployed models remain appropriate as market conditions evolve. Financial institutions typically maintain multiple calibration models running in parallel, with consistency checks comparing results across methods to detect potential failures or market regime changes that might invalidate model assumptions. Neural network approaches fit naturally into such frameworks as one component of a diverse methodology toolkit rather than as complete replacements for traditional methods.

The training data requirements for neural calibration systems deserve particular attention since model performance depends critically on covering the parameter space appropriately during training. Historical market data provides valuable information about parameter ranges actually observed in practice, enabling training datasets that concentrate probability mass in high-relevance regions rather than spreading uniformly across theoretically possible values. However, relying exclusively on historical observations risks inadequate coverage of extreme scenarios that might occur during market stress, precisely when accurate calibration matters most for risk management. Balancing historical realism against robustness to outliers through mixture distributions combining observed parameter distributions with broader support proves essential for production reliability.

The validation and monitoring of deployed neural calibration systems requires ongoing attention as market conditions evolve. Automated systems should continuously compare neural calibration results against traditional methods on representative subsets of calibrations, flagging instances where discrepancies exceed tolerance thresholds for manual review. Metrics tracking the distribution of calibrated parameters over time can identify gradual drift suggesting model degradation requiring retraining or architectural modifications. The frequency of retraining depends on market characteristics, with volatile environments exhibiting frequent regime changes potentially requiring monthly or quarterly retraining while stable markets might maintain accuracy over longer horizons. However, the offline nature of training means retraining costs typically prove acceptable given the accumulated value from thousands of rapid calibrations between training cycles.

Regulatory and compliance considerations increasingly shape the adoption of machine learning methods in financial applications. Regulators have expressed concerns about black-box models whose decision logic remains opaque, potentially obscuring risks or enabling manipulation. Neural calibration systems can partially address these concerns through careful documentation of training data, architecture choices, and validation procedures, combined with ongoing comparison against traditional methods providing interpretable parameter estimates. Some institutions implement neural methods primarily as pricing accelerators within traditional optimization frameworks rather than as standalone calibration systems, maintaining transparency by using established algorithms for parameter selection while leveraging neural approximations only for rapid pricing evaluation during optimization iterations.

The integration of neural calibration with broader quantitative infrastructure including pricing libraries, risk systems, and trading platforms requires careful software engineering. Modern production systems typically employ microservice architectures where calibration services expose standardized interfaces accepting market data and returning calibrated parameters, with the internal calibration methodology abstracted behind this interface. This design enables gradual migration from traditional to neural methods, with production systems initially running both approaches in parallel for validation before gradually shifting traffic to neural implementations as confidence builds. Containerization and orchestration technologies facilitate deploying multiple model versions simultaneously, enabling A-B testing and gradual rollout strategies that minimize disruption risk during method transitions.

5 CONCLUSION

This comprehensive investigation of deep learning approaches to stochastic volatility model calibration establishes both the substantial practical advantages and remaining theoretical challenges associated with neural network methods in quantitative finance applications. The analysis demonstrates that careful attention to numerical issues in characteristic function evaluation, particularly branch-switching discontinuities that corrupt pricing calculations, proves essential for achieving reliable calibration regardless of whether traditional optimization or neural approximation methods are employed. The detailed examination of neural network architectures incorporating exponential linear unit activations and deep hierarchical representations reveals how modern deep learning frameworks can effectively approximate the complex nonlinear mappings connecting model parameters to option prices, enabling dramatic computational acceleration while maintaining accuracy sufficient for production applications.

The systematic empirical comparison across genetic algorithms, adaptive simulated annealing, nonlinear least squares optimization, and various neural network configurations provides quantitative evidence that gradient-based local optimization methods substantially outperform global stochastic search algorithms for stochastic volatility calibration when properly initialized. The lsqnonlin algorithm consistently achieved average absolute relative errors below one

percent across diverse objective function weightings, establishing a clear benchmark for neural approaches to match or exceed. The observation that different algorithms recover substantially different parameter values despite achieving similar aggregate pricing errors highlights fundamental challenges in calibration where objective function structure permits multiple solutions that prove equivalent for calibration options but diverge for out-of-sample applications, suggesting that neural network validation must extend beyond simple prediction error assessment to include parameter stability analysis.

Several important limitations of current methodologies warrant acknowledgment and motivate future research directions. The black-box nature of neural networks creates challenges for interpretability in an industry where understanding model behavior under stress scenarios and explaining decisions to regulators remains paramount. While neural networks demonstrate impressive interpolation within training data distributions, their extrapolation behavior outside these ranges proves less predictable than parametric models with established theoretical properties. The substantial initial investment required for training neural networks, particularly when incorporating sophisticated architectures and comprehensive training datasets, represents a barrier to adoption compared to traditional methods immediately applicable without offline training phases, though this cost amortizes over many subsequent calibrations.

Future research should prioritize developing neural architectures that more explicitly incorporate financial domain knowledge, such as no-arbitrage constraints, asymptotic pricing behaviors, and relationships between different maturities and strikes that financial theory establishes. Physics-informed neural networks that embed known differential equation structure into architectures through specialized layers or loss function terms represent particularly promising directions for improving both accuracy and interpretability. Investigation of uncertainty quantification methods providing confidence intervals for calibrated parameters rather than point estimates would address critical gaps in current neural approaches, enabling more principled risk management decisions that account for calibration uncertainty. Extension beyond vanilla stochastic volatility to more complex specifications including jumps, stochastic interest rates, and multiple volatility factors represents important application domains where neural methods' flexibility and speed advantages over traditional approaches may prove even more compelling.

From a practical implementation perspective, financial institutions should consider adopting neural network calibration through carefully phased deployment strategies that initially run neural methods in parallel with traditional approaches for validation before gradually transitioning production traffic. This risk-mitigation approach enables building institutional confidence in neural methods while preserving traditional calibration as fallback when neural predictions appear unreliable. The implementation should maintain flexibility to update training datasets and retrain models as market conditions evolve, with monitoring systems tracking calibration quality and flagging potential degradation requiring model updates. Investment in robust computational infrastructure supporting rapid inference, version control, and comprehensive logging proves essential for reliable production deployment.

The broader implications of this research extend beyond immediate calibration applications to fundamental questions about the role of machine learning in quantitative finance. The success of neural networks in approximating expensive pricing functions suggests similar approaches might prove valuable for other computational bottlenecks including Monte Carlo simulation, partial differential equation solvers, and Greeks calculations. However, the observation that different calibration methods achieving similar in-sample errors can produce substantially different parameter estimates emphasizes that purely data-driven approaches without appropriate domain knowledge incorporation risk missing important structure. The optimal path forward likely involves hybrid methodologies that combine domain-specific modeling assumptions with flexible machine learning components, leveraging complementary strengths rather than viewing these paradigms as competing alternatives.

In conclusion, deep learning approaches to stochastic volatility model calibration represent significant methodological advances offering clear practical benefits for computational efficiency and robustness, though they do not eliminate fundamental challenges inherent in inferring unobservable parameters from market prices. The careful characterization of numerical challenges in characteristic function evaluation and the detailed analysis of neural architectures and optimization algorithms provided in this work offer valuable guidance for researchers and practitioners implementing next-generation calibration systems. As the methodology matures and best practices emerge, neural network calibration seems likely to become a standard component of the quantitative analyst's toolkit, complementing rather than replacing traditional methods and enabling more sophisticated modeling with faster adaptation to evolving market conditions than previously possible.

COMPETING INTERESTS

The authors have no relevant financial or non-financial interests to disclose.

REFERENCES

[1] Nafiu A, Balogun S O, Oko-Odion C, et al. Risk management strategies: Navigating volatility in complex financial market environments. World Journal of Advanced Research and Reviews, 2025, 25(1): 236-250.

[2] Ma Z, Chen X, Sun T, et al. Blockchain-based zero-trust supply chain security integrated with deep reinforcement learning for inventory optimization. Future Internet, 2024, 16(5): 163.

[3] Sun T, Yang J, Li J, et al. Enhancing auto insurance risk evaluation with transformer and SHAP. IEEE Access, 2024.

[4] Patawari A, Das P. From traditional to computationally efficient scientific computing algorithms in option pricing: Current progresses with future directions. Archives of Computational Methods in Engineering, 2025: 1-40.
[5] Li Q, Jiang H, Lee C H. Phase-Space Generalized Brillouin Zone For Spatially Inhomogeneous Non-Hermitian Systems. Advanced Science, 2025: e08047.
[6] Zhang Y, Zhang H. Enhancing robot path planning through a twin-reinforced chimp optimization algorithm and evolutionary programming algorithm. IEEE Access, 2025, 12: 170057-170078.
[7] Gilli M, Maringer D, Schumann E. Numerical methods and optimization in finance. Academic Press, 2019.
[8] Devan M, Thirunavukkarasu K, Shanmugam L. Algorithmic trading strategies: real-time data analytics with machine learning. Journal of Knowledge Learning and Science Technology, 2023, 2(3): 522-546.
[9] Khalili H. Spatial competition of learning agents in agricultural procurement markets. Doctoral dissertation, Universitäts-und Landesbibliothek Bonn, 2019.
[10] Nikolenko S I. Synthetic data for deep learning. Cham, Switzerland: Springer, 2021, 174.
[11] Ge Y, Wang Y, Liu J, et al. GAN-Enhanced Implied Volatility Surface Reconstruction for Option Pricing Error Mitigation. IEEE Access, 2025.
[12] Hirsa A. Computational methods in finance. Chapman and Hall/CRC, 2024.
[13] Sabbar Y. A Review of Modern Stochastic Modeling: SDE/SPDE Numerics, Data-Driven Identification, and Generative Methods with Applications in Biology and Epidemiology. arXiv preprint arXiv:2508.11004, 2025.
[14] Shlezinger N, Whang J, Eldar Y C, et al. Model-based deep learning. Proceedings of the IEEE, 2023, 111(5): 465-499.
[15] Rohilla A. Exploring Volatility: Evolution, Advancements, Trends, and Applications. 2024.
[16] Alòs E, León J A. An intuitive introduction to fractional and rough volatilities. Mathematics, 2021, 9(9): 994.
[17] Bajaj P, Kaur J. Recent Developments and Applicability of the Black and Scholes Model in Option Pricing: A Literature Review. Mudra: Journal of Finance & Accounting, 2020, 7(2).
[18] Kristensen S. High-Performing Computational Methods for American Option Pricing and Implied Volatility Surface Construction. 2025.
[19] Cao W, Mai N T, Liu W. Adaptive knowledge assessment via symmetric hierarchical Bayesian neural networks with graph symmetry-aware concept dependencies. Symmetry, 2025, 17(8): 1332.
[20] Mai N T, Cao W, Liu W. Interpretable knowledge tracing via transformer-Bayesian hybrid networks: Learning temporal dependencies and causal structures in educational data. Applied Sciences, 2025, 15(17): 9605.
[21] Liu S, Borovykh A, Grzelak L A, et al. A neural network-based framework for financial model calibration. Journal of Mathematics in Industry, 2019, 9(1): 9.
[22] Chen S, Liu Y, Zhang Q, et al. Multi-Distance Spatial-Temporal Graph Neural Network for Anomaly Detection in Blockchain Transactions. Advanced Intelligent Systems, 2025: 2400898.
[23] Zhang Q, Chen S, Liu W. Balanced Knowledge Transfer in MTTL-ClinicalBERT: A Symmetrical Multi-Task Learning Framework for Clinical Text Classification. Symmetry, 2025, 17(6): 823.
[24] Ren S, Jin J, Niu G, et al. ARCS: Adaptive Reinforcement Learning Framework for Automated Cybersecurity Incident Response Strategy Optimization. Applied Sciences, 2025, 15(2): 951.
[25] Moradinia Z, Vandierendonck H, Murphy A. Navigating speed–accuracy trade-offs for multi-physics simulations. Meccanica, 2025, 60(6): 1583-1599.
[26] Mai N T, Cao W, Wang Y. The global belonging support framework: Enhancing equity and access for international graduate students. Journal of International Students, 2025, 15(9): 141-160.
[27] Liu Y, Ren S, Wang X, et al. Temporal logical attention network for log-based anomaly detection in distributed systems. Sensors, 2024, 24(24): 7949.
[28] Tan Y, Wu B, Cao J, et al. LLaMA-UTP: Knowledge-Guided Expert Mixture for Analyzing Uncertain Tax Positions. IEEE Access, 2025.
[29] Zheng W, Liu W. Symmetry-Aware Transformers for Asymmetric Causal Discovery in Financial Time Series. Symmetry, 2025, 17(10): 1591.
[30] Ivaşcu C F. Option pricing using machine learning. Expert Systems with Applications, 2021, 163: 113799.
[31] Hu X, Zhao X, Wang J, et al. Information-theoretic multi-scale geometric pre-training for enhanced molecular property prediction. PLoS One, 2025, 20(10): e0332640.
[32] Zhang H, Ge Y, Zhao X, et al. Hierarchical deep reinforcement learning for multi-objective integrated circuit physical layout optimization with congestion-aware reward shaping. IEEE Access, 2025.
[33] Wang J, Zhang H, Wu B, et al. Symmetry-Guided Electric Vehicles Energy Consumption Optimization Based on Driver Behavior and Environmental Factors: A Reinforcement Learning Approach. Symmetry, 17(6), 930.
[34] Hu, X., Zhao, X., & Liu, W. (2025). Hierarchical Sensing Framework for Polymer Degradation Monitoring: A Physics-Constrained Reinforcement Learning Framework for Programmable Material Discovery. Sensors, 2025, 25(14): 4479.
[35] Wang Y, Ding G, Zeng Z, et al. Causal-Aware Multimodal Transformer for Supply Chain Demand Forecasting: Integrating Text, Time Series, and Satellite Imagery. IEEE Access, 2025.
[36] Han X, Yang Y, Chen J, et al. Symmetry-Aware Credit Risk Modeling: A Deep Learning Framework Exploiting Financial Data Balance and Invariance. Symmetry, 2025, 17(3): 20738994.