Efficient Pricing of Exotic Derivatives:

Deep Learning Approximation under GBM, Heston and Bates Models

Project Head:

Michael **Ladaa**

Team Members:

Aleksandar **Georgiev**
Mattia **Arecco**
Neil **Maaouni**
Raffaele **Simbari**
Salomé **Messika**

HEPHAESTUS APPLIED ARTIFICIAL INTELLIGENCE ASSOCIATION

Fall 2025

PROJECT REPORT

Abstract

This report investigates the approximation of exotic derivative pricing functionals using deep neural networks trained on Monte Carlo simulated data. We construct a large synthetic dataset under Geometric Brownian Motion, the Heston stochastic volatility model, and the Bates jump-diffusion model, and evaluate the accuracy and robustness of neural network pricing approximations across multiple path-dependent contracts.

Keywords: exotic derivatives, Monte Carlo simulation, stochastic volatility, jump-diffusion, neural networks.

Contents

1 Introduction

Exotic derivatives play a central role in modern financial markets, offering tailored risk exposures that cannot be replicated with standard European-style contracts. In contrast to general options, exotic derivatives often depend on the entire price path of the underlying asset, incorporate state-dependent features such as barriers, or embed nonlinear payoff structures. While these characteristics enhance flexibility in risk management and structured product design, they substantially increase pricing complexity. In many cases, closed-form valuation formulas are unavailable, especially under realistic asset dynamics such as stochastic volatility or jump-diffusion processes.

Monte Carlo simulation provides a flexible and model-consistent framework for valuing such contracts. Under correct model specification, Monte Carlo estimators are unbiased and converge at rate $O(N^{-1/2})$, making them a natural benchmark for complex derivative pricing. However, high accuracy requires a large number of simulated paths, rendering Monte Carlo methods computationally expensive. This computational burden becomes particularly restrictive in real-time applications such as intraday risk monitoring, portfolio optimization, calibration, or scenario analysis, where thousands of prices must be computed repeatedly.

Recent advances in machine learning offer a promising alternative. Neural networks can approximate high-dimensional nonlinear functions with remarkable accuracy and speed once trained. In the context of derivative pricing, the mapping from model parameters and contract characteristics to arbitrage-free prices can be interpreted as a deterministic functional

$$f(\theta) = \mathbb{E}^{\mathbb{Q}}\left[e^{-rT}\Phi(S_{[0,T]};\theta)\right],$$

which is continuous but highly nonlinear. By training on a sufficiently rich set of Monte Carlo-generated price observations, a neural network can learn this pricing functional and provide near-instantaneous predictions, effectively amortizing the computational cost of simulation.

The objective of this paper is therefore twofold. First, we construct a large synthetic dataset of exotic derivative prices generated under three increasingly sophisticated asset dynamics: Geometric Brownian Motion (GBM), the Heston stochastic volatility model, and the Bates stochastic volatility jump-diffusion model. This hierarchy allows us to systematically analyze how model complexity affects learnability. Second, we evaluate the ability of fully-connected neural networks to approximate Monte Carlo benchmark prices across different payoff structures, including arithmetic Asian options, barrier options, and floating-strike lookback options.

By separating models and payoff types into independent supervised learning tasks, we investigate how smooth versus discontinuous pricing functionals influence approximation quality. Our results show that neural networks approximate smoother payoffs such as Asian and lookback options with high accuracy, while barrier contracts and jump-diffusion dynamics remain significantly more challenging. These findings provide insight into the interaction between model complexity, payoff discontinuities, and neural-network approximation capacity.

Overall, this study contributes to the growing literature on deep learning in quantitative finance by providing a controlled experimental comparison across asset dynamics and exotic payoff structures, highlighting both the strengths and limitations of standard feedforward architectures in derivative pricing tasks.

2 Data Generation

2.1 Model Hierarchy and Simulation Framework

The dataset is constructed under three increasingly sophisticated asset price models: Geometric Brownian Motion (GBM), the Heston stochastic volatility model, and the Bates stochastic volatility jump-diffusion model.

These models form a natural hierarchy of structural complexity:

- **GBM** assumes constant volatility and continuous paths.
- **Heston** introduces stochastic variance with mean reversion.
- **Bates** augments Heston by adding discontinuous jump components.

This layered structure ensures that the neural network is exposed to progressively richer dynamics, ranging from purely diffusive behavior to stochastic volatility with rare extreme events.

All simulations are performed under the risk-neutral probability measure $\mathbb{Q}$, ensuring that discounted asset prices satisfy the martingale condition

$$\mathbb{E}^{\mathbb{Q}}\left[e^{-rT}S_T\right] = S_0.$$

Monte Carlo simulation is used to approximate the pricing functional

$$f(\theta) = \mathbb{E}^{\mathbb{Q}}\left[e^{-rT}\Phi(S_{[0,T]};\theta)\right],$$

where θ denotes the parameter vector of the respective model and Φ the option payoff functional.

2.2 Random Walk and Brownian Motion

To explain the Brownian motion process and its mechanics, we begin with a clear definition of a random walk. A random walk is a stochastic process in which, at each discrete time step, the state of the system changes by a random increment. In its simplest symmetric one-dimensional form, the process moves up or down by a fixed step with equal probability.

A random walk is naturally a martingale. Its conditional expectation given the current information set equals its present value, and the unconditional expectation remains zero over time. Furthermore, the variance of the random walk increases proportionally with the number of time steps, reflecting the accumulation of uncertainty over time. As the time discretisation becomes finer and the step size appropriately rescaled, the random walk converges in distribution to Brownian motion. This provides a conceptual bridge between discrete-time stochastic processes and continuous-time models.

A standard Brownian motion (also called a Wiener process) $(W_t)_{t\geq 0}$ is a continuous-time stochastic process characterized by the following properties:

- $W_0 = 0$,
- Independent increments: for $0 \leq s < t$, the increment $W_t - W_s$ is independent of $\mathcal{F}_s$,
- Normally distributed increments:

$$W_t - W_s \sim \mathcal{N}(0, t - s). \tag{1}$$

An important implication of this property is that the variance of the increment equals the elapsed time interval:

$$\mathrm{Var}(W_t - W_s) = t - s.$$

Brownian motion has continuous sample paths and no jumps, although the paths are highly irregular and nowhere differentiable.

Brownian motion itself does not incorporate a systematic trend and therefore cannot describe financial asset prices, which exhibit both drift and randomness.

2.2.1 Arithmetic Brownian Motion

A simple extension introduces a deterministic trend component, yielding the Arithmetic Brownian Motion (ABM). A process X_t follows ABM if it satisfies the stochastic differential equation

$$dX_t = \mu\, dt + \sigma\, dW_t, \tag{2}$$

where μ denotes the constant drift parameter and σ the diffusion coefficient.

The ABM describes dynamics in which the expected change over time is linear while random fluctuations are added independently. However, this model allows negative values, which makes it unsuitable for modeling stock prices. Moreover, fluctuations are additive rather than proportional to the current value.

2.2.2 Geometric Brownian Motion

To obtain a more realistic model for asset price dynamics, we introduce Geometric Brownian Motion (GBM). Under GBM, both drift and diffusion scale with the current value of the process. The asset price process S_t satisfies

$$dS_t = \mu S_t\, dt + \sigma S_t\, dW_t. \tag{3}$$

Thus, the relative change in asset price is stochastic rather than the absolute change. The process remains strictly positive almost surely, and the magnitude of fluctuations scales with the current price level.

Applying Itô's Lemma to the logarithm of the asset price $X_t = \log(S_t)$ yields

$$dX_t = \left(\mu - \frac{1}{2}\sigma^2\right) dt + \sigma dW_t. \tag{4}$$

Hence, the log-price follows an arithmetic Brownian motion. Consequently, log-returns are normally distributed and asset prices follow a log-normal distribution. This property is central to the Black–Scholes framework and provides a convenient probabilistic structure for analytical option pricing and Monte Carlo simulation.

The GBM admits the explicit solution

$$S_t = S_0 \exp\left(\left(\mu - \frac{1}{2}\sigma^2\right) t + \sigma W_t\right). \tag{5}$$

This exponential structure forms the foundation of the classical Black–Scholes option pricing framework. In this project, GBM serves as a baseline model for generating synthetic asset price paths, which are subsequently used to compute exotic option payoffs and to construct training datasets for neural-network-based pricing models. Peter 2023

2.3 The Heston Stochastic Volatility Model

The constant-volatility assumption of the Geometric Brownian Motion is often too restrictive to capture empirical properties of financial markets, such as volatility clustering, heavy tails, and volatility smiles. To relax this assumption, Heston (1993) introduced a stochastic volatility framework in which the asset price is modeled jointly with a time-varying variance process.

Let S_t denote the asset price, r the constant risk-free rate, and δ the constant dividend yield. Under the risk-neutral measure, the Heston dynamics are given by

$$dS_t = (r - \delta)S_t\,dt + S_t\sqrt{V_t}\,dZ_t, \tag{6}$$

$$dV_t = \kappa\,(\theta - V_t)\,dt + \xi\sqrt{V_t}\,dW_t. \tag{7}$$

Here, V_t denotes the instantaneous variance rate. The variance process follows a square-root diffusion of Cox–Ingersoll–Ross (CIR) type. It is mean-reverting: the parameter $\kappa > 0$ determines the speed at which variance is pulled back toward its long-run level $\theta > 0$. The parameter $\xi > 0$ represents the *volatility of volatility* and controls the magnitude of stochastic fluctuations in variance. The initial variance is denoted by $V_0 = v_0$.

The two Wiener processes Z_t and W_t are correlated:

$$dZ_t\,dW_t = \rho\,dt, \tag{8}$$

where $\rho \in [-1, 1]$ captures the instantaneous correlation between asset returns and variance innovations. This feature allows the model to reproduce the empirically observed leverage effect, namely the negative correlation between returns and volatility changes.

In contrast to the GBM framework, where volatility is constant and log-returns are fully characterized by a single parameter σ, the Heston model introduces a second source of randomness through the variance process V_t. As a consequence, return distributions become conditionally Gaussian but unconditionally non-Gaussian, enabling heavier tails and more flexible implied-volatility structures.

The Heston model therefore provides a substantially richer and more realistic description of asset price dynamics, while still retaining semi-analytical tractability for European-style derivatives.

2.4 The Bates Stochastic Volatility Jump-Diffusion Model

2.4.1 Motivation

While the Black-Scholes framework assumes constant volatility and continuous asset price paths, empirical financial data strongly contradicts these assumptions. Asset returns exhibit volatility clustering, heavy tails, skewness, and occasional abrupt price movements. Stochastic volatility models, such as the Heston model Heston 1993, address time-varying volatility and leverage effects but still assume continuous price trajectories. However, real markets frequently experience sudden jumps caused by macroeconomic announcements, geopolitical events, or liquidity shocks. To incorporate both stochastic volatility and discontinuous price movements, we employ the Bates model Bates 1996, which augments the Heston stochastic volatility framework with lognormal jumps in the asset price. This hybrid specification combines diffusion-driven fluctuations with rare but significant jump events and is therefore particularly suitable for modeling exotic derivatives that are sensitive to tail risk and barrier breaches.

2.4.2 Model Dynamics

Under the risk-neutral probability measure $\mathbb{Q}$, the Bates model is defined by the following system of stochastic differential equations:

$$dS_t = rS_t dt + \sqrt{v_t} S_t dW_t^S + S_{t^-}(J-1)dN_t, \tag{9}$$
$$dv_t = \kappa(\theta - v_t)dt + \xi\sqrt{v_t}dW_t^v, \tag{10}$$

with instantaneous correlation

$$dW_t^S dW_t^v = \rho dt.$$

Here, S_t denotes the asset price and v_t the instantaneous variance process. The variance follows a Cox–Ingersoll–Ross (CIR) type mean-reverting process with long-run level θ, mean-reversion speed κ, and volatility-of-volatility parameter ξ. The Brownian motions W_t^S and W_t^v are correlated with coefficient ρ, allowing the model to reproduce leverage effects observed in equity markets.

The jump component is modeled by a Poisson process N_t with intensity λ, representing the arrival rate of jumps. Jump sizes are lognormally distributed:

$$\log J \sim \mathcal{N}(\mu_J, \delta_J^2).$$

Thus, when a jump occurs, the asset price is multiplied by the random factor J. This specification corresponds to the jump-diffusion model originally introduced by Merton Merton 1976, embedded within a stochastic volatility environment.

To preserve the martingale property under $\mathbb{Q}$, the drift is adjusted by the jump compensator term $-\lambda\mathbb{E}[J-1]$. This ensures that discounted asset prices remain martingales and derivative prices are arbitrage-free.

2.4.3 Economic Interpretation

The Bates model captures three key empirical features of financial markets.

First, stochastic variance allows for persistent periods of high or low volatility, reflecting volatility clustering. Second, negative correlation between returns and variance innovations generates asymmetric return distributions and implied volatility skews. Third, the jump component introduces sudden large price movements that generate heavy tails and excess kurtosis in return distributions.

These features are particularly relevant for exotic derivatives. Barrier options, for example, are highly sensitive to sudden large price movements that may trigger knock-out events. Lookback options depend on path extrema and therefore respond strongly to jump-induced spikes. Ignoring jumps in such contexts may lead to systematic mispricing.

2.4.4 Numerical Simulation

Closed-form solutions for most exotic derivatives under the Bates model are unavailable. Consequently, Monte Carlo simulation is required.

The simulation procedure combines two components:

1. Simulation of the stochastic volatility diffusion using the Quadratic-Exponential (QE) scheme of Andersen Andersen 2008a, which preserves non-negativity of the variance process and ensures numerical stability.
2. Simulation of Poisson-distributed jump arrivals at each time step, followed by generation of normally distributed log-jump sizes.

Given a time discretization Δt, the number of jumps within each interval is drawn from

$$N_{\Delta t} \sim \text{Poisson}(\lambda \Delta t).$$

Conditional on $N_{\Delta t} = k$, the cumulative log-jump is normally distributed with mean $k\mu_J$ and variance $k\delta_J^2$. The asset price update therefore consists of two multiplicative components: the continuous diffusion term and the exponential jump term.
This separation allows efficient vectorized simulation and preserves consistency with the underlying continuous-time specification.

2.4.5 Implications for Deep Learning Approximation

The inclusion of jumps significantly increases the complexity of the pricing functional

$$f(\theta) = \mathbb{E}^{\mathbb{Q}}\left[e^{-rT}\Phi(S_{[0,T]};\theta)\right].$$

Relative to pure diffusion models, the presence of jumps introduces stronger nonlinearities and heavier tails in the payoff distribution. For barrier and lookback options, even a single jump can dramatically alter the realized payoff.
From a machine learning perspective, this enriches the training dataset and improves the robustness of the neural network approximation. The network is forced to learn pricing behavior not only under smooth diffusive dynamics but also in the presence of discontinuities and rare extreme events.
Consequently, the Bates model serves as a demanding benchmark for evaluating the approximation capacity of deep neural networks in exotic derivative pricing tasks.

2.5 Parameter Sampling and Numerical Stability

Model parameters are randomly sampled from economically meaningful ranges. The sampling procedure ensures sufficient coverage of the relevant pricing domain while maintaining numerical stability of the simulation scheme.

For GBM, the parameter vector is

$$\theta_{\text{GBM}} = (S_0, K, r, T, \sigma).$$

For Heston, the parameter vector becomes

$$\theta_{\text{HESTON}} = (S_0, K, r, T, v_0, \kappa, \theta, \xi, \rho),$$

where:

- v_0 is the initial variance,
- κ the mean-reversion speed,
- θ the long-run variance,
- ξ the volatility of volatility,
- ρ the correlation between return and variance shocks.

For the Bates model, jump parameters are added:

$$\theta_{\text{BATES}} = \theta_{\text{HESTON}} \cup (\lambda, \mu_J, \delta_J),$$

where:

- λ is the jump intensity,
- μ_J the mean log-jump size,
- δ_J the jump size volatility.

For barrier options, an additional parameter B is included:

$$B = \text{barrier\_mult} \cdot S_0.$$

This relative parametrization ensures economically realistic barrier placements.

2.6 Numerical Simulation Schemes

The simulation procedure depends on the selected model:

- **GBM:** Simulated using exact log-Euler discretization,

$$S_{t+\Delta t} = S_t \exp\left(\left(r - \frac{1}{2}\sigma^2\right)\Delta t + \sigma\sqrt{\Delta t}Z\right).$$

- **Heston:** Variance paths are simulated using the Quadratic-Exponential (QE) scheme of Andersen 2008b, which preserves non-negativity and improves numerical stability.
- **Bates:** Combines the QE diffusion scheme with Poisson-distributed jump arrivals:

$$N_{\Delta t} \sim \text{Poisson}(\lambda \Delta t).$$

Antithetic variates are employed to reduce variance of Monte Carlo estimators. Simulation is performed in memory-controlled batches to allow large-scale dataset generation.

2.7 Exotic Derivatives Considered

The dataset comprises three major classes of path-dependent derivatives: arithmetic Asian options, barrier options, and floating-strike lookback options. Unlike standard European options, whose payoff depends solely on the terminal asset price S_T, these contracts depend on the entire trajectory $\{S_t\}_{t\in[0,T]}$. Path dependence significantly increases pricing complexity and generally eliminates the availability of closed-form valuation formulas under realistic asset dynamics.

Arithmetic Asian Options

Asian options are contracts whose payoff depends on an average of the underlying asset price over a predetermined set of monitoring dates $\{t_1, \dots, t_n\}$. The primary motivation behind Asian options is risk reduction through averaging, which mitigates the impact of short-term price manipulation or extreme terminal price realizations Kemna and Vorst 1990.

In the case of an arithmetic Asian call, the payoff is given by

$$\Phi_{\text{Asian Call}} = \max\left(\frac{1}{n}\sum_{i=1}^{n} S_{t_i} - K, 0\right).$$

Similarly, the arithmetic Asian put payoff is

$$\Phi_{\text{Asian Put}} = \max\left(K - \frac{1}{n}\sum_{i=1}^{n} S_{t_i}, 0\right).$$

Because the arithmetic average of lognormal variables is not lognormal, closed-form pricing formulas generally do not exist, even under the Black–Scholes framework. Under stochastic volatility or jump-diffusion dynamics, valuation must therefore rely on numerical methods such as Monte Carlo simulation Glasserman 2004.

Economically, Asian options are widely used in commodity and energy markets, where average price exposure over a delivery period is more relevant than spot price risk at maturity.

Barrier Options

Barrier options are contingent claims whose existence or payoff depends on whether the underlying asset price crosses a predetermined barrier level B during the option's lifetime Hull 2018. These contracts are classified as either knock-in or knock-out options.

In this study, we consider knock-out structures. For example, the payoff of an up-and-out call option is defined as

$$\Phi = \begin{cases} \max(S_T - K, 0), & \text{if } \max_{t\leq T} S_t < B, \\ 0, & \text{if } \max_{t\leq T} S_t \geq B. \end{cases}$$

The option becomes worthless if the barrier is breached at any time before maturity. This introduces a discontinuous payoff structure and strong path dependence.

Barrier options are typically cheaper than their vanilla counterparts because the additional barrier condition reduces the probability of payout. However, they are highly sensitive to volatility and jump risk: even a single extreme price movement may activate or deactivate the contract. Under jump-diffusion dynamics, this sensitivity becomes particularly pronounced, making barrier options an informative test case for model risk and numerical approximation accuracy Broadie, Glasserman, and Kou 1997.

Floating-Strike Lookback Options

Lookback options allow the holder to retrospectively select the most favorable price observed during the contract's lifetime. These contracts depend on the extrema of the price path and therefore exhibit extreme path dependence Goldman, Sosin, and Gatto 1979.

For a floating-strike lookback call, the payoff is

$$\Phi_{\text{Lookback Call}} = \max\left(S_T - \min_{t\leq T} S_t, 0\right).$$

For a floating-strike lookback put,

$$\Phi_{\text{Lookback Put}} = \max\left(\max_{t\leq T} S_t - S_T, 0\right).$$

Unlike fixed-strike options, the strike is determined endogenously by the realized path. This feature makes lookback options particularly valuable in volatile markets, as they effectively insure against adverse timing of exercise.

Because extrema of stochastic processes are involved, pricing formulas become analytically intricate even under geometric Brownian motion and typically require advanced probabilistic techniques. Under stochastic volatility or jump-diffusion models, Monte Carlo simulation becomes the primary valuation tool.

3 The Neural network

3.1 Neural Network Setup

After generating a synthetic dataset of exotic options priced via Monte Carlo, we trained a NN to approximate the pricing function. Traditional pricing of exotic instruments via Monte Carlo (MC) simulations is computationally expensive and slow for real-time risk management. For this reason, we train a NN to approximate the pricing function $f(\text{parameters}) \to \text{price}$.

A key design choice in our implementation is to train separate networks for each (model, option) pair rather than training a single network. This is motivated by the fact that different payoff families behave very differently (e.g. , Asian depends on averages, Lookback depends on extrema, Barriers introduce threshold effects), and a single model tends to struggle and gives not reliable results when it must handle heterogeneous behaviors at once.

Once trained, the model provides pricing faster, enabling instantaneous calculation and high-frequency risk monitoring compared to Monte Carlo estimation.

3.2 Neural Network inputs per model

3.2.1 GBM Input Parameters

For the GBM subset, we filter the dataset to retain only contracts generated under the GBM dynamics. For each option type, we select the target column price and build the input feature vector as follows:

- **Basic contract inputs:** S_0, K, r, T.
- **GBM volatility parameter:** σ.
- **Barrier level:** Included only when the option type is a barrier product (e.g., B).

3.2.2 Heston Input Parameters

For the Heston experiments, each network takes as input the following feature vector:

- **Base contract inputs:** S_0, K, r, T.
- **Heston parameters:** $v_0, \kappa, \theta, \xi, \rho$.
- **Option-specific parameter:** Barrier level B (included only for barrier option types).

3.2.3 Bates Input Parameters

For the Bates experiments, each network takes as input the following features:

- **Base contract inputs:** S_0, K, r, T.

- **Stochastic volatility parameters:** $v_0, \kappa, \theta, \xi, \rho$.
- **Jump parameters:** $\lambda_j, \mu_j, \delta_j$.
- **Option-specific parameter:** Barrier level B (included only for barrier option types).

3.3 Model pipeline (common to all experiments)

For a given (model, option) pair, we performed data selection and feature construction, input features are standardized using the mean and standard deviation computed on the training set only, and the same transformation is applied to validation.
Training is performed with a fully-connected feedforward neural network (GELU activations, Layer Normalization and Dropout for regularization), optimized using MSE loss and AdamW.
To ensure robustness, we repeat training over multiple random seeds and report mean and standard deviation of validation metrics; for diagnostics, we also save predicted vs. true prices for the best-performing seed in order to generate Predicted-vs-True plots.

3.4 Architecture and model training

Given the model-specific input vectors defined above, each (model, option) pair is treated as an independent supervised regression task, where the input is the selected parameter vector and the target is the corresponding Monte Carlo price. Consistently with the project design, we train our network per (model, option) pair to avoid forcing a single model to capture heterogeneous payoff structures and parameters domains.

Technical Architecture The network is a fully connected regressor with five linear layers (four hidden layers plus a final linear output layer). Every hidden layer applies:
Linear $\rightarrow$ GELU $\rightarrow$ LayerNorm $\rightarrow$ Dropout,
with hidden width 128 and dropout probability 0.10. The output layer is a single linear neuron, producing a scalar price estimate. Only the input dimension varies across models/options according to the parameters defined above. All models were implemented in PyTorch.

Training protocol The training protocol consists of using MSE loss with the AdamW optimizer (learning rate 10^{-3}, weight decay 10^{-4}). A `ReduceLROnPlateau` scheduler (factor 0.5, patience 5) adapts the learning rate when validation loss stalls, and gradients are clipped to a maximum norm of 5.0 for stability. Data are split into 70% training / 30% validation, and early stopping is applied on validation MSE (min_delta $= 10^{-4}$, patience $= 15$, maximum epoch budget $= 300$); the best checkpoint is restored at the end of the training. That is, training stops after 300 epochs, or earlier if the validation MSE does not improve by at least 10^{-4} for 15 consecutive epochs.

Finally, we repeat the training over 10 random seeds and report mean and standard deviation of validation metrics across runs.

3.5 Model Evaluation

After training, the model parameters are restored from the checkpoint achieving the lowest validation MSE. Final metrics are then computed on both the training set and the validation set. Let $\{(x_i, y_i)\}_{i=1}^{N}$ denote a dataset (train or validation) and $\hat{y}_i = f_\theta(x_i)$ the corresponding predictions. On both datasets we report:

- **Mean Squared Error (MSE):**

$$\text{MSE} = \frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i)^2. \tag{11}$$

- **Root Mean Squared Error (RMSE):**

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i)^2} = \sqrt{\text{MSE}}. \tag{12}$$

- **Mean Absolute Error (MAE):**

$$\text{MAE} = \frac{1}{N}\sum_{i=1}^{N}|\hat{y}_i - y_i|. \tag{13}$$

In addition, on the validation set only, we compute two tolerance-based coverage metrics. With $\varepsilon = 10^{-8}$, an absolute tolerance $\tau_{\text{abs}} = 0.05$, and a relative tolerance $\tau_{\text{rel}} = 0.20$, we report:

- **Absolute tolerance coverage:**

$$\text{Cov}_{\text{abs}}(\tau_{\text{abs}}) = \frac{1}{N}\sum_{i=1}^{N}\mathbf{1}\left(|\hat{y}_i - y_i| \le \tau_{\text{abs}}\right). \tag{14}$$

- **Relative tolerance coverage:**

$$\text{Cov}_{\text{rel}}(\tau_{\text{rel}}) = \frac{1}{N}\sum_{i=1}^{N}\mathbf{1}\left(\frac{|\hat{y}_i - y_i|}{|y_i| + \varepsilon} \le \tau_{\text{rel}}\right). \tag{15}$$

For simplicity and readability, in the following result tables we report only **MAE**, **RMSE**, and the **relative tolerance coverage** $\text{Cov}_{\text{rel}}(\tau_{\text{rel}})$.

4 Results

Across the three dynamics (GBM, Heston, and Bates), the network achieves its best average performance on Asian arithmetic and floating-strike lookback options, while barrier options remain the most difficult class. Under GBM, Asian and lookback contracts yield relatively low validation errors (e.g., validation RMSE around 1.41–3.03 and validation MAE around 0.65–1.22), and the gap between train and validation metrics is moderate. Under Heston, the same payoff families show higher, but still controlled, validation errors (validation RMSE roughly 2.6–5.1, validation MAE roughly 1.5–3.0), consistent with the increased complexity of stochastic-volatility pricing.

Barrier options display substantially worse average metrics across models. In GBM, barrier contracts show larger validation losses and very low relative tolerance coverage at the 20% threshold. In Heston, barrier options are again among the worst performers, with particularly high errors for the up-and-out call. This pattern is consistent with the fact that barrier payoffs induce sharper, less smooth dependence on inputs, which is harder to approximate accurately with standard regression training.

Moving from GBM to Heston and especially to Bates, performance degrades across all payoffs. The Bates results show the highest validation errors for every option type reported: for instance, validation RMSE is approximately 3.2–6.4 for Asian/lookback options and exceeds 4 for the barrier cases, with correspondingly low absolute-tolerance coverage (0.01–0.21). Overall, the average metrics indicate that the jump-diffusion setting introduces additional complexity that is not captured as effectively by the current fully-connected baseline and feature set.

Option	Split	MAE	RMSE	Rel. cov.
ASIAN_ARITH_CALL	Train	0.58	1.24	0.75
ASIAN_ARITH_PUT	Train	0.54	1.28	0.75
BARRIER_DOWN_AND_OUT_PUT	Train	0.44	1.91	0.03
BARRIER_UP_AND_OUT_CALL	Train	0.60	2.99	0.11
LOOKBACK_FLOAT_CALL	Train	0.61	1.21	0.98
LOOKBACK_FLOAT_PUT	Train	0.58	1.22	0.98
ASIAN_ARITH_CALL	Val	0.70	1.35	0.74
ASIAN_ARITH_PUT	Val	0.66	1.40	0.73
BARRIER_DOWN_AND_OUT_PUT	Val	0.65	2.13	0.03
BARRIER_UP_AND_OUT_CALL	Val	1.22	4.02	0.10
LOOKBACK_FLOAT_CALL	Val	0.68	1.33	0.98
LOOKBACK_FLOAT_PUT	Val	0.65	1.41	0.98

Table 1: Mean metrics over seeds for GBM (train vs validation).

Option	Split	MAE	RMSE	Rel. cov.
ASIAN_ARITH_CALL	Train	0.95	2.24	0.65
ASIAN_ARITH_PUT	Train	0.82	1.86	0.74
BARRIER_DOWN_AND_OUT_PUT	Train	0.54	2.05	0.05
BARRIER_UP_AND_OUT_CALL	Train	0.88	4.31	0.11
LOOKBACK_FLOAT_CALL	Train	0.89	1.66	0.98
LOOKBACK_FLOAT_PUT	Train	0.86	1.65	0.99
ASIAN_ARITH_CALL	Val	1.56	2.84	0.60
ASIAN_ARITH_PUT	Val	1.48	2.56	0.70
BARRIER_DOWN_AND_OUT_PUT	Val	1.35	3.29	0.05
BARRIER_UP_AND_OUT_CALL	Val	3.05	6.61	0.10
LOOKBACK_FLOAT_CALL	Val	1.62	2.66	0.97
LOOKBACK_FLOAT_PUT	Val	1.48	2.66	0.97

Table 2: Mean metrics over seeds for Heston (train vs validation).

Option	Split	MAE	RMSE	Rel. cov.
ASIAN_ARITH_CALL	Train	1.00	2.45	0.69
ASIAN_ARITH_PUT	Train	1.04	2.30	0.73
BARRIER_DOWN_AND_OUT_PUT	Train	0.60	2.52	0.04
BARRIER_UP_AND_OUT_CALL	Train	0.73	3.17	0.10
LOOKBACK_FLOAT_CALL	Train	1.07	1.87	0.98
LOOKBACK_FLOAT_PUT	Train	1.05	1.88	0.99
ASIAN_ARITH_CALL	Val	1.99	3.45	0.58
ASIAN_ARITH_PUT	Val	2.10	3.60	0.63
BARRIER_DOWN_AND_OUT_PUT	Val	1.85	4.13	0.01
BARRIER_UP_AND_OUT_CALL	Val	3.89	7.84	0.04
LOOKBACK_FLOAT_CALL	Val	2.03	3.20	0.21
LOOKBACK_FLOAT_PUT	Val	2.02	3.24	0.18

Table 3: Mean metrics over seeds for Bates (train vs validation).

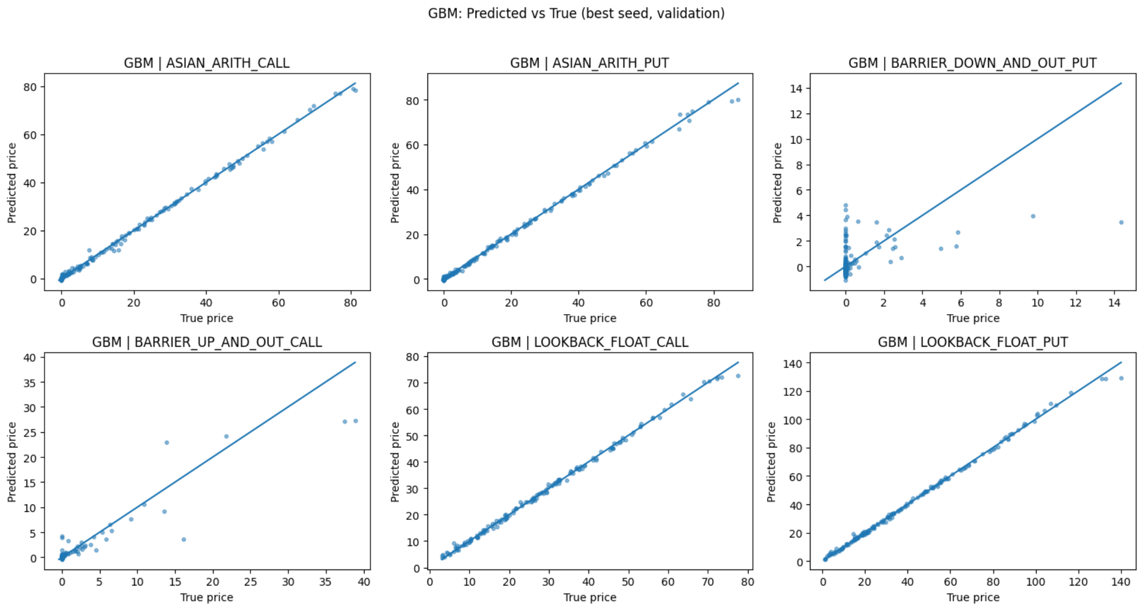

Figure 1: GBM: Predicted vs True (best seed, validation).

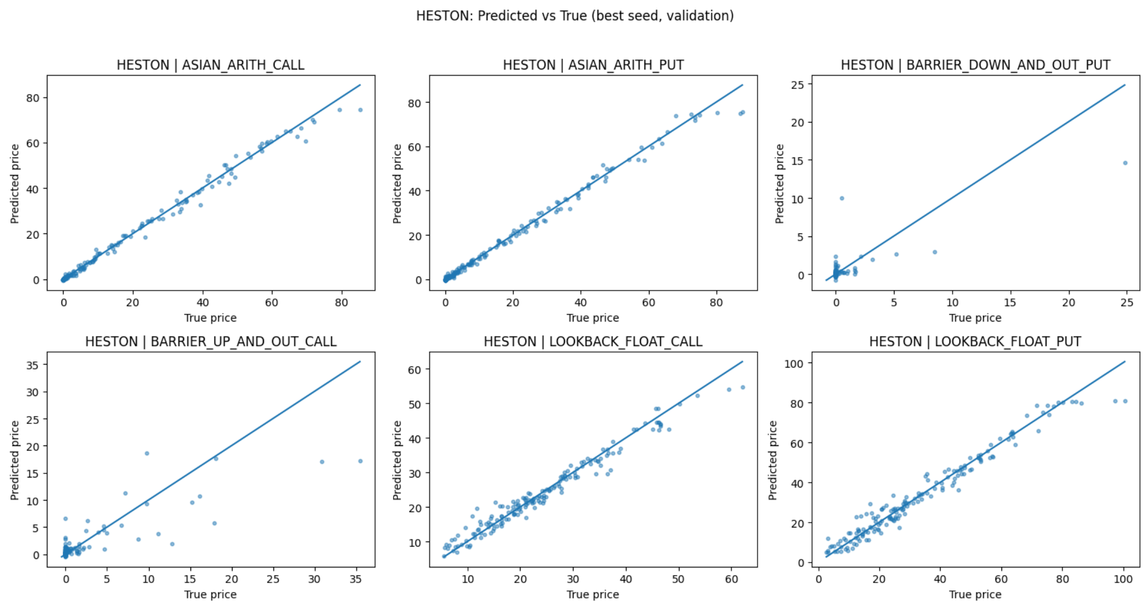

Figure 2: Heston: Predicted vs True (best seed, validation).

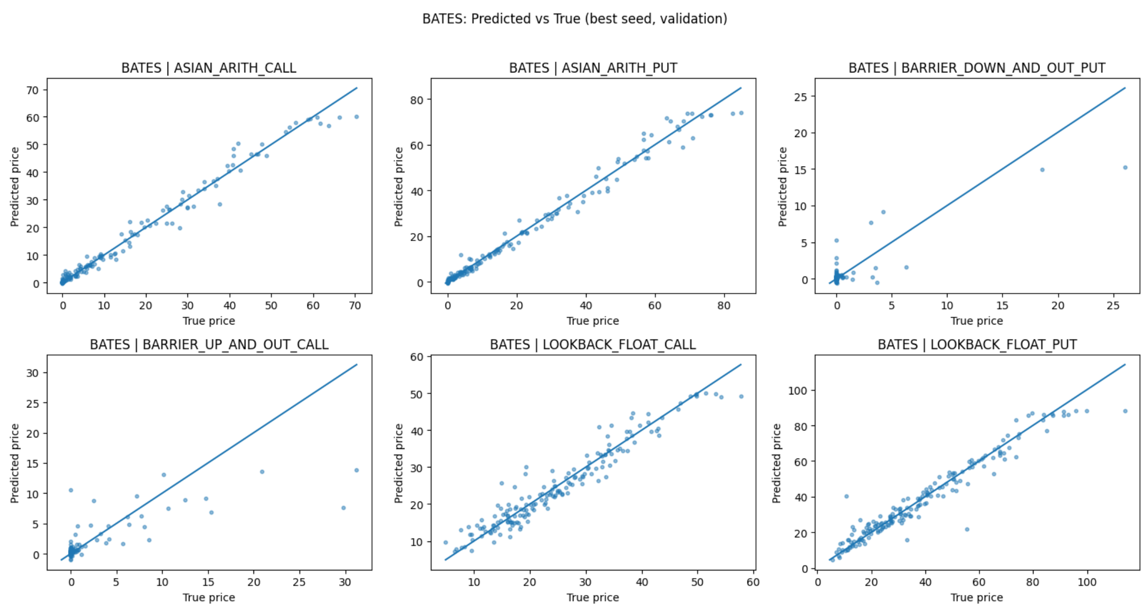

Figure 3: Bates: Predicted vs True (best seed, validation).

5 Conclusion

Overall, the experiments show that a simple fully-connected regressor trained per (model,option)(model,option)(model,option) pair can approximate Monte Carlo prices reasonably well for smoother exotic payoffs, while it struggles on contracts with sharper threshold effects and on more complex dynamics. Under GBM, Asian arithmetic and floating-strike lookback options achieve low average validation errors (e.g., validation RMSE $\approx 1.4 - 3.0$ and validation MAE $\approx 0.65 - 1.22$), indicating that the pricing functional for these payoffs is learned effectively from the available synthetic data. Under

Heston, errors increase but remain moderate for the same payoff families (validation RMSE $\approx 2.6 - 5.1$), suggesting that the network still captures a substantial portion of the additional nonlinearity introduced by stochastic volatility. In contrast, barrier options are consistently the hardest cases across models, with substantially larger validation losses and very low relative-tolerance coverage, highlighting that knock-out features create a more discontinuous mapping from parameters to prices.

A clear next step is to improve data coverage and model specialization in the regimes where performance deteriorates (barriers and the Bates jump-diffusion setting). In the current setup, the synthetic dataset contains only about 10,000 samples total, which are then split across roughly 18 separate networks (one per (model,option) pair). This implies that each network is trained on a relatively small subset of the data (often only a few hundred observations), limiting generalization-especially for payoffs that are rare or highly nonlinear (e.g., barrier contracts). A straightforward improvement is therefore to increase the size of the synthetic dataset substantially, with particular attention to sampling more densely around 'critical' regions (such as parameters that place the underlying close to the barrier). Complementary modeling improvements include (i) a two-stage approach for barriers (first predicting knock-out/near-zero behavior, then regressing the conditional price) and/or (ii) weighted losses that emphasize barrier-near samples. Finally, regime-specific diagnostics (near vs. far from the barrier; low vs. high jump intensity) would help localize the dominant error sources and guide targeted data generation and architecture choices.

References

Andersen, Leif B. G. (2008a). "Simple and Efficient Simulation of the Heston Stochastic Volatility Model". In: *Journal of Computational Finance* 11.3, pp. 1–42.

– (2008b). "Simple and Efficient Simulation of the Heston Stochastic Volatility Model". In: *Journal of Computational Finance* 11.3, pp. 1–42. DOI: 10.21314/JCF.2008.189.

Bates, David S. (1996). "Jumps and Stochastic Volatility: Exchange Rate Processes Implicit in Deutsche Mark Options". In: *Review of Financial Studies* 9.1, pp. 69–107.

Broadie, Mark, Paul Glasserman, and Steven G. Kou (1997). "A Continuity Correction for Discrete Barrier Options". In: *Mathematical Finance* 7.4, pp. 325–349. DOI: 10.1111/1467-9965.00035.

Glasserman, Paul (2004). *Monte Carlo Methods in Financial Engineering*. Vol. 53. Applications of Mathematics: Stochastic Modelling and Applied Probability. New York, NY: Springer Science+Business Media. ISBN: 978-0-387-00451-8. DOI: 10.1007/978-0-387-21617-1.

Goldman, Michael B., Howard B. Sosin, and Mary Ann Gatto (1979). "Path Dependent Options: "Buy at the Low, Sell at the High"". In: *Journal of Finance* 34.5, pp. 1111–1127. DOI: 10.1111/j.1540-6261.1979.tb00067.x.

Heston, Steven L. (1993). "A Closed-Form Solution for Options with Stochastic Volatility". In: *Review of Financial Studies* 6.2, pp. 327–343.

Hull, John C. (2018). *Options, Futures, and Other Derivatives*. 10th ed. Pearson.

Kemna, Angelien G. Z. and A. C. F. Vorst (1990). "A Pricing Method for Options Based on Average Asset Values". In: *Journal of Banking and Finance* 14.1, pp. 113–129. DOI: 10.1016/0378-4266(90)90046-8.

Merton, Robert C. (1976). "Option Pricing when Underlying Stock Returns are Discontinuous". In: *Journal of Financial Economics* 3.1-2, pp. 125–144.

Peter, Milan (2023). *From a Coin-Toss to the Greeks: Building the Black-Scholes Model Step by Step with Mathematical and Visual Insights*. Independently published. ISBN: 979-8395042846.