Making the Heston Model More Practical with Deep Learning

Joshua Quintieri
Erick Antonio Martinez Sevilla
Mustafa Ark
Nicolas Hidalgo
Mirza Aydemir

Machine Learning in Finance (MATH 6912)

Dr. Michael Chen

York University

April 19, 2026

Making the Heston Model More Practical with Deep Learning

Introduction and Motivation

To price options for managing risk, creating trading strategies and optimizing portfolios successfully, accurate pricing of equity options on a global basis is critical. Option pricing models have typically been traditional, such as the Black-Scholes Option Pricing Model, which assumes the underlying asset(s) have a constant volatility (and all other inputs) and a continuous path. The options market is subject to many empirically derived characteristics regarding price behaviour, including volatility clusters, skewness (or asymmetry) of the distribution curve and a volatility smile, which are not taken into consideration within these models.

Consequently, the development of stochastic option pricing models, such as the Heston Model, is imperative in order to gain an accurate picture of the price relationships that exist in the real options market. The Heston Model utilises the concept of stochastic volatility, resulting in a non-constant time dependent volatility of the underlying asset(s).

Even though it has been shown to be more accurate than other models, utilisation of the Heston Model for actual trading purposes presents challenges due to the calibration being very computationally intense; it is highly sensitive to noise in the data; and relatively unstable as a result of non-linear parameter space in the Heston Model. As a result, it has been very difficult to effectively use the Heston Model for actual trading.

As a means to address these challenges, this proposed project seeks to integrate traditional financial modelling methodologies with deep learning methodologies (machine learning) for the purpose of creating a hybrid financial modelling framework which retains the Heston Model interpretability while improving calibration speed, robustness and accuracy. By combining structured financial theory with neural network approximations, the project aims to create a more practical and scalable option pricing pipeline.

Black-Scholes vs Heston Model

The Black-Scholes option pricing model assumes constant volatility but is very easy to use. As a result, it has become the most widely used option price model in finance. The trade off for using this simple and easy-to-use option pricing model is it cannot explain the volatility smile, which occurs when different implied volatilities are assigned to deep in-the-money and out-of-the-money options.

The lack of validity of the Black-Scholes option pricing model to account for the fact that market data shows different implied volatilities for each type of option contradicts the assumption behind the Black-Scholes pricing model.

The Heston option pricing model improves on the Black-Scholes model by adding a second stochastic process to capture the variance of options. In other words, the variance of the option is no longer assumed constant, but rather is modelled as a mean-reverting process that is characterized by the following parameters:

* Mean reversion speed (k)

* Long-term variance (θ)

* Volatility of volatility (σ)

* Correlation of returns on the underlying asset and volatility (ρ)

* Initial variance (v0).

Assuming that the underlying assumptions of the Heston model hold true, it captures:

* Volatility smiles and skews

* Time-varying volatility

* The leverage effect (negative correlation between returns and volatility)

Although examining whether the Heston model provides a good fit to actual market data demonstrates that the Heston model does provide some improvements over the Black-Scholes model, it is also worth noting that the Heston model does not have a simple closed-form solution, and requires the use of numerical or approximate methods to price options. Furthermore, the calibration of the Heston model requires repeated option pricing calculations, making it computationally cumbersome, and can be highly dependent on the noise in the data.

This project aims to improve Heston's computational efficiency via machine learning techniques while retaining its core mathematical framework.

Framework Overview

To overcome the limitations of traditional calibration, the project introduces a structured pipeline that integrates financial modeling with neural networks. The framework consists of the following phases, each addressing a specific challenge:

1. Synthetic Data Generation
   A large dataset of Heston parameters is generated using a Latin Hypercube Generator, which is then used to obtain a synthetic option surface using the Heston semi-closed form solution. The synthetic data covers a wide range of parameters that will allow for optimized pricing.
2. Surrogate Network
   Instead of computing Heston prices using slow numerical methods, a neural network is trained to approximate the pricing function. This dramatically reduces computation time during calibration.
3. Price Approximator Network (PAN)
   Real market data is often noisy due to bid-ask spreads and illiquid trades. PAN smooths this data by learning a clean relationship between strike prices and option prices. This ensures that calibration focuses on underlying market structure rather than noise.
4. Initial Heston Calibration
   Using the surrogate model and PAN-smoothed prices, the framework estimates Heston parameters by minimizing pricing errors.
5. Calibration Correction Network (CCN)
   Since the Heston model cannot perfectly fit all market features, a neural network is used to learn and correct residual pricing errors.

This pipeline combines:

- Financial interpretability (Heston parameters)
- Computational efficiency (surrogate model)
- Noise reduction (PAN)
- Accuracy improvement (CCN)

As described in the report, this layered approach ensures that each component adds value, resulting in a more accurate and practical pricing system

Surrogate Network

The surrogate network replaces the numerical Heston pricing function during calibration. This helps calibration by decreasing the time spent and computing power needed to calibrate the model by skipping closed form solution which requires nearly a second to calibrate each option in an optimization that consists of thousands of options. The surrogate network maps 10 inputs of the option to the option price and only takes microseconds to evaluate each option.

While this is not a part of the original paper, we wanted to add it to speed up calibration given the limited computing power and time.

Training Data

We trained the surrogate on the 1.6 million synthetic samples generated earlier. It takes 10 inputs: spot price S_0, strike K, log-moneyness m, time to maturity T, mean reversion speed κ, long-term variance θ, volatility of volatility σ, correlation ρ, initial variance v_0, and the risk-free rate r.
The output is the Heston price computed by the Euler-Maruyama scheme. We used a standard 90/10 train-test split and scaled all features and targets.

Architecture

We initially used a smaller network with layers of 10, 24, 16, and 1 neurons, but it could be improved to capture the full Heston pricing surface across all parameter regions. The final architecture has four layers: 10 inputs → 128 (ReLU) → 64 (ReLU) → 32 (ReLU) → 1 output. Training used the Adam optimizer with a learning rate of 0.001.

And we set early stopping with a patience of 5 epochs to stop training when no further improvement is observed. The maximum number of epochs was 30.

Results

The surrogate achieved a best validation MSE of 0.000005, which translates to an RMSE of approximately 2.19 dollars across the full parameter range. This level of accuracy is sufficient for our use case. Because the surrogate sits inside an optimization loop, small approximation errors are handled by the optimizer as it searches across the parameters. The calibration process will work as long as the surrogate network maintains the general shape of the pricing surface.

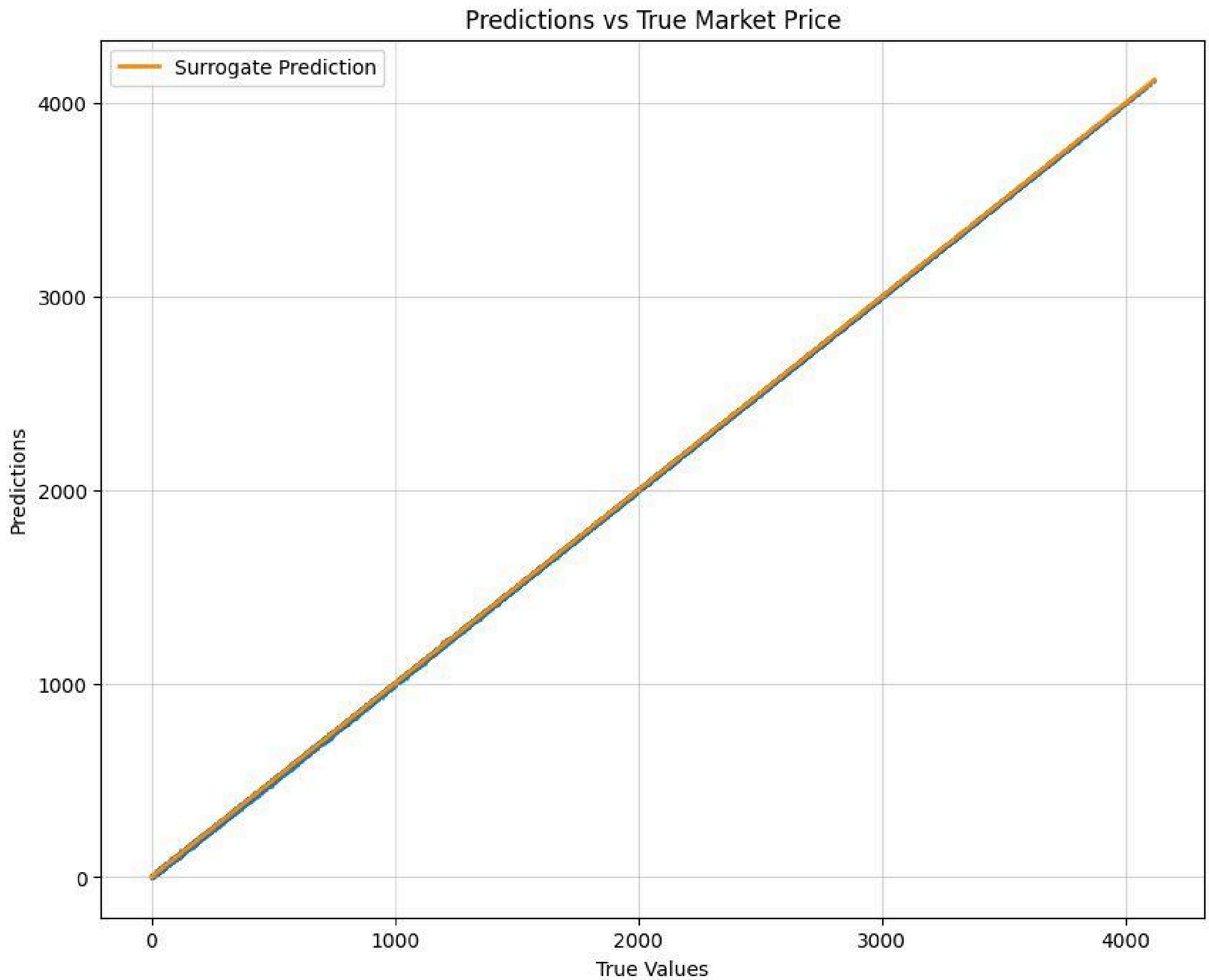


Figure 1: Surrogate Network Predictions vs. True Heston Option Prices

With the surrogate trained and validated, we now have a fast pricing function ready for use during calibration. However, before calibrating, we need to ensure we have clean real market data to calibrate against.

Data Collection and Preparation

Quality of data was of highest importance since Heston Calibration is very sensitive to noisy or inaccurate market data. The model's non-linear parameter space also meant small errors in input prices can lead to large errors in calibrated parameters. Thus data cleaning became essential before calibration.

Our S&P 500 call option data was pulled from Yahoo Finance through the yfinance library in Python. The ticker ^SPX provided option chain data including strike price, last price, bid, ask, volume, open interest and implied volatility. We obtained Spot price using ^GSPC. Near term expirations selected to reflect the result of the papers accurately.

The following are the steps we took in order to clean the data:

1. Duplicate removal: Duplicate strikes have been removed.
2. Implied Volatility floor has been set.
3. Outlying strike prices have been removed.

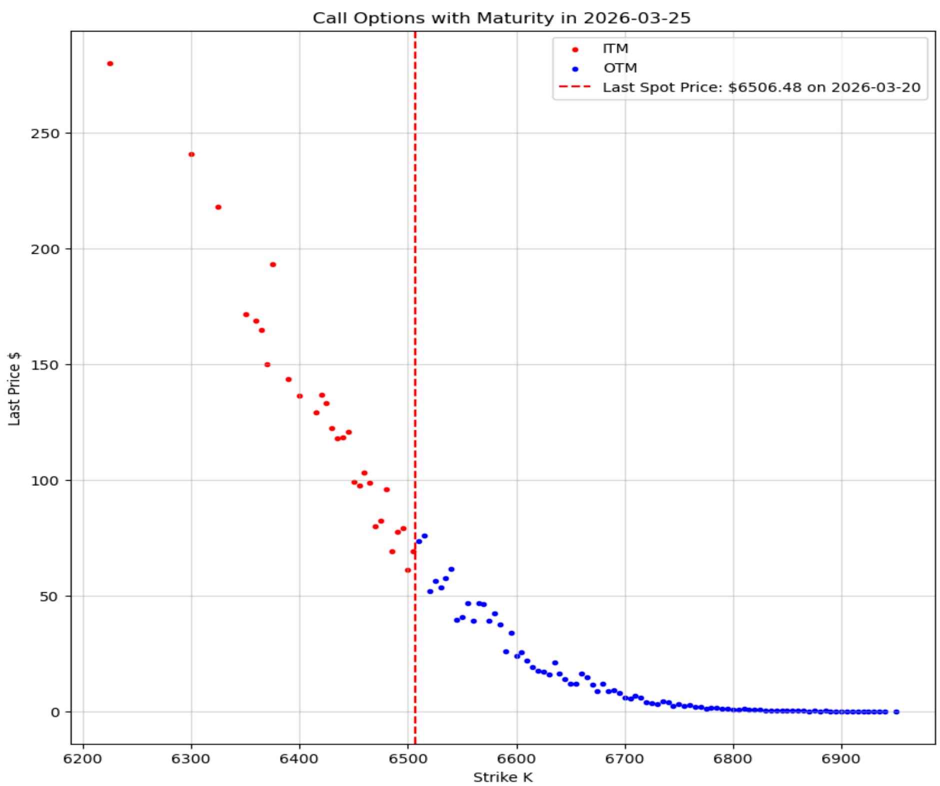


Figure 2: Cleaned Market Call Option Prices Across Strike Prices for the 2026-03-25 Maturity

The cleaned data set has been saved to use effectively in Price Approximator Network (PAN) for training and calibration.

Price Approximator Network (PAN)

The PAN learns a smooth relationship from strike price K to option price P for a single maturity. This is essential since even after the data-cleaning, market prices contain some noise from bid-ask spread and low volume trades. PAN however provides a smooth price surface that calibration optimizer can use.

Architecture is a very classic definition of a feedforward network that consists of four layers: 1 input (Strike Price) → 8 (Tanh) → 8 (ReLU) → 1 output (Option Price). The first hidden layer uses the Tanh activation function which centers data around zero with output range between -1 and 1. This stabilizes early gradients. The second hidden layer uses ReLU activation function which fixes the vanishing gradient problem and efficiently learns the non-linear pricing curve. The output layer is linear that enables the PAN to predict any price value. In total there will be 97 trainable parameters.
The PAN also uses Xavier initialization for Tanh and Kaiming initialization for ReLU.

While this architecture is small, it is very sufficient since there are only 120 market observations and the task is not complicated enough to require more neurons. We also avoid overfitting through early stopping conditions.

The training set up consists of 90% of training data and 10% of testing data. Both inputs and price targets are standardized with scaler. Adam optimizer used with the earliest stopping condition of 30 epochs with a maximum number of 500 epochs set.

Training converged with early stopping at epoch 253. The best validation loss achieved was 0.250036, with a final learning rate of 2.50e-04.

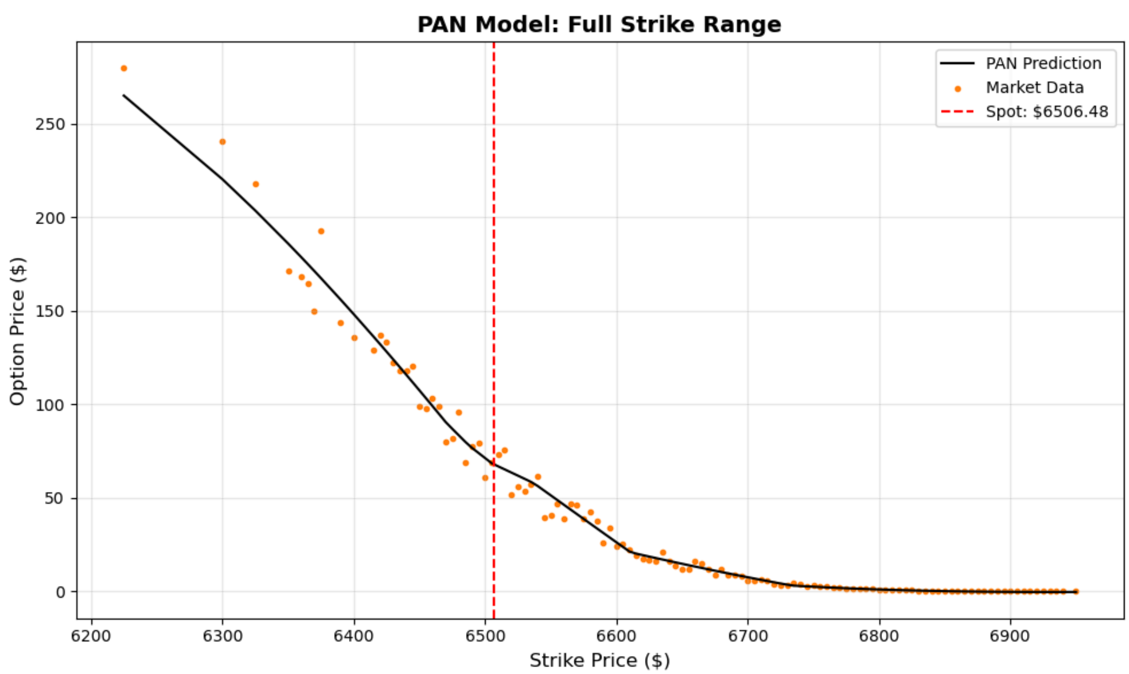


Figure 3: PAN Prediction with Strike Range

The plot shows the PAN curve passing through the center of the market data without chasing individual outlier points. It descends steeply on the ITM side and flattens toward zero on the

OTM side, capturing the characteristic shape of a call option pricing curve while ignoring the scattered data points that represent noise rather than signal.

Using the PAN smoothed prices instead of raw market prices makes the optimization surface smoother and avoids the optimizer fitting noise instead of actual data.

Initial Heston Calibration

After training the surrogate model and the PAN network, we moved on to the initial Heston calibration. Here we start estimating actual Heston parameters from market based information instead of just approximating prices. Previously, the PAN model gave us a smooth version of market prices depending on strikes, and the surrogate model gave a fast way to generate Heston prices for various parameter sets.
The goal is to find the following Heston parameters that makes the surrogate network based Heston prices as close as possible to the PAN smoothed market prices: Kappa, theta, sigma, rho, v0.

- Kappa controls how fast variance moves back to its long-run level.
- Theta measures the long-run variance level.
- Sigma, the volatility of variance, often called vol-of-vol.
- Rho shows the correlation between the stock-price shock and the variance shock.
- v0, the initial variance.

In the code we do this inside the *calibration_heston()* function. For each strike price, this function builds the surrogate input using the spot price, strike price, moneyness, maturity, the five Heston parameters we mentioned earlier, and the risk free rate. It then passes the calculated inputs to the trained surrogate network, converts the prediction back to price space, normalizes it by spot, and compares it to the normalized PAN price. The objective function calculates model prices minus market prices for every strike, then squares those differences, adds them all up, and divides by the total number of options which gives us the mean squared error (MSE), which is what the optimizer tries to make as small as possible.
Before the optimization begins, we provide an initial parameter guess to guide the search. In our calibration run, the initial parameter vector was:
[kappa, theta, sigma, rho,v0] = [2, 0.06, 0.51, -0.7, 0.059].
These are not the final calibrated values. They are only the starting point from which the optimizer begins searching for a better-fitting parameter set.
It is important to note that we are not calibrating with raw market prices directly, instead we use PAN smoothed prices. This is done because the raw option data we collected can be noisy especially in the less liquid strikes. Using this setup, the optimization converged successfully after 157 iterations and 290 function evaluations. The final fit produced an MSE of 0.000002 and an RMSE of 0.001471.

These results give us the first full Heston parameter set implied by our calibration framework. One thing worth mentioning is that some of the calibrated parameters ended up at the edge of the allowed region. This suggests that the optimizer is pushing the basic Heston model as far as it can in order to fit the data. So even though the optimization was successful and the pricing error is very small, the result also shows that the plain Heston model still has some difficulty fully matching the market surface on its own. We also checked the Feller condition (2*kappa*theta > sigma^2) and determined that the condition is not satisfied.

Overall, this step gives us the first calibrated Heston parameter set based on the PAN smoothed market surface. It is a crucial step because it turns the framework into a real calibration system with economically interpretable parameters. At the same time, the boundary solutions and Feller violation suggest that the initial Heston fit is still not enough by itself. This is exactly why we need the next stage, the Calibration Correction Network (CCN), which is designed to learn the remaining pricing errors that the initial calibrated Heston model cannot fully capture.

Calibration Correction Network (CCN)

After the initial Heston calibration, we still found that the model was not fully matching the market prices. Even though the initial fit was better than a plain uncalibrated model, some pricing errors were still left especially for strikes where the market surface is harder to capture with the standard Heston structure. This is why we introduced the Calibration Correction Network (CCN). CCN's role is to take prices coming from the calibrated Heston model and learn how to adjust them so they move closer to the observed market prices.

The main idea behind CCN is that the initial calibration already captures a large part of market behavior, but not everything. Some of the remaining errors are not just random noise. They are more systematic which means that the calibrated Heston model may consistently misprice certain parts of the option surface. Instead of recalibrating the full model again, CCN learns this leftover correction directly. It works like a final adjustment layer on top of the initial Heston fit.

In our code, the CCN is built as a quite simple one input, one output neural network. The input is the price that was produced by the initially calibrated Heston model and the target is the market price that we want the network to learn. We have prepared this training data in the *data_prep_ccn()* function, where the feature is the Heston price and the target is either the market price. The network is not trying to relearn the whole option pricing problem. It is only learning the correction from the initial Heston fit to the target price surface.

The CCN architecture takes one input then passes it to two hidden layers, and returns one corrected output price. Based on the final project setup, the first hidden layer has 7 neurons with a sigmoid activation, the second hidden layer has 7 neurons with a tanh activation, and the output layer gives the final corrected price.

For training, the data is split into training and test sets, scaled using standardization, and then passed into *PyTorch* dataloaders. The model is trained using the Adam optimizer and MSE loss. A learning-rate scheduler is also used during training, and early stopping is triggered if the validation loss stops improving. In our code network is allowed to train for up to 1000 epochs, and the best model is saved during training which was used later for evaluation. This part is important because it helps prevent overfitting and makes sure we keep the version of the network that performs best on unseen data.

Once training is complete, the corrected outputs are converted back into the original price scale and compared against the market data. The final CCN results show that the correction step improves the pricing surface by reducing the gap between model prices and observed option prices. The stabilized model results after applying CCN are a Real RMSE of $5.93, Real MAE of $4.74, and Real MRE of 69.68%. Even if some error still remains, this stage clearly improves the fit by correcting systematic pricing mistakes that the initial calibrated Heston model could not remove by itself.

This section is important because it shows that the project does not stop at standard Heston calibration. The initial calibrated model gives us a good starting point, but it still has structural limits. The CCN improves the final fit without throwing away the financial meaning of the Heston parameters. So the Heston model still provides the main structure, while the neural network only learns the remaining adjustment needed to better match the market.

The CCN acts as the final refinement step in the single maturity pipeline. It takes the output from the initially calibrated Heston model, learns the leftover pricing error, and produces a corrected price surface that is closer to the market. This is what makes the full framework stronger than a standard calibration only approach. It combines the structure and interpretability of Heston with the flexibility of a neural network to capture what the model misses.

Full Pipeline Results and Single Maturity Model Comparison

Here we show what each stage actually contributed. The goal here is to compare the pricing performance of three versions of the model, the traditional Heston model, the initially calibrated Heston model, and the final CCN corrected model. Looking at all three together lets us see not only whether the model improved, but also where each step of the pipeline added value gradually.

We start with the traditional Heston model as the baseline. This gives us a reference point before applying the NN based improvements. In the baseline case, the Heston model is able to capture the general shape of the option price curve across strikes, especially the downward pattern as strike increases. It also tends to fit better around the at-the-money region. However, the fit becomes weaker in other parts of the surface, especially in the tails. In particular, the model has difficulty matching some deep in-the-money prices and can underprice/overprice options further away from the spot region. This is the main limitation of relying on the plain Heston structure alone.

The next stage is the initial calibrated Heston model. This already improves the fit because the parameters are now estimated using the PAN smoothed market surface rather than being fixed arbitrarily. Also, instead of relying on slow repeated Heston pricing, the calibration uses the surrogate network to make the optimization practical. As we saw in the previous section, this step produced a very small fitting error during calibration, with the optimizer converging successfully. So compared to the traditional Heston model, the initial calibrated version gives a better fit to the target surface and serves as a stronger starting point for the final correction stage.

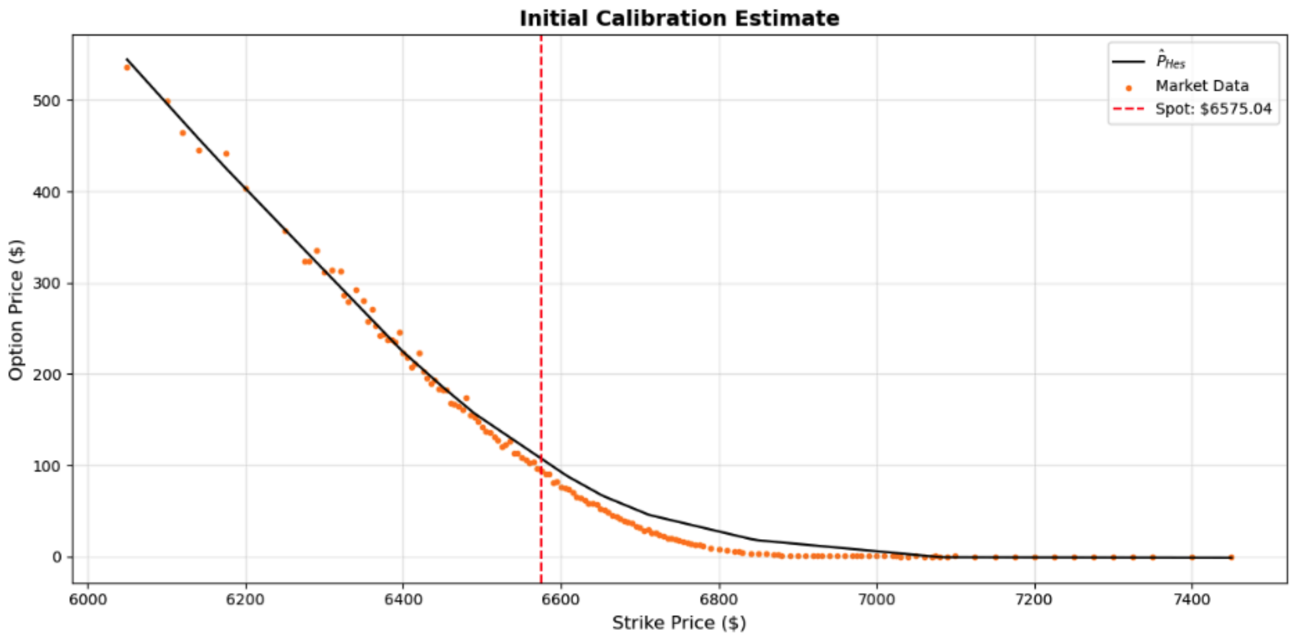


Figure 4: Initial Heston Calibration Fit Compared with Observed Market Option Prices Across Strike Prices

Even with this improvement, the initial calibrated Heston model still does not fully match the observed market prices. Some of the remaining pricing errors are systematic. This is where the CCN becomes important. Instead of recalibrating the structural model again, CCN learns the remaining correction directly from the initially calibrated Heston prices. This gives us the final version of the pipeline: a calibrated Heston model with an additional learned correction layer on top.

After applying CCN, the final fitted curve moves closer to the market data and better captures the shape of the single-maturity option surface. This is the main result of the full framework. The structural part of the model still comes from Heston, so the parameters remain economically interpretable, but the neural network helps fix the pricing errors that Heston alone cannot remove.

Figure 5: Comparison of Market Option Prices, Initial Heston Calibration, and CCN-Corrected Predictions Across Strike Prices

The full pipeline is able to combine financial structure with machine learning flexibility. The Heston model still gives a meaningful volatility framework, while the surrogate model, PAN, and CCN make the method faster and more accurate. Another strength of this pipeline is that each stage has a clear role. The surrogate model speeds up pricing, PAN smooths the noisy target surface, calibration estimates the structural parameters, and CCN fixes the remaining pricing

error. The single-maturity results show that the full pipeline performs better than a standard Heston-only approach.

Extension to Multiple Maturities

Here we are focusing on interpreting the calibrated Heston parameters across different maturities to understand their financial meaning and how market expectations evolve over time.

The original framework was extended from a single maturity to multiple maturity setting by combining option data from several expiration dates into one unified dataset. For each option, the pipeline computes T, the time to maturity. This is coupled with finding the log-moneyness upon which all maturities are appended to the same table before training the PAN. This allows the model to learn a pricing surface that depends on how far the contract is from expiration. In the calibration stage, the Heston parameters are fit using strikes, maturities, risk-free rates, and PAN smoothed prices across all options.

This extension makes the framework more realistic because real option markets are quoted across many maturities, not just one.

When looking at changes across a 3, 6, and 12 month calibration, it is important to note that the main pipeline structure stays the same. It takes into account data preparation, PAN training, Heston calibration, CCN correction, and then evaluation. The steps are done in this order to maintain the most accurate results.

The changes across 3, 6, 12-month calibration is the input maturity information and the associated risk-free rate for each contract. The helper code explicitly builds maturity specific T values and estimates corresponding risk-free rates for different expiration dates by using an interpolation method on the treasury yield curve. The PAN looks at features such as strike, time to maturity, and log-moneyness. This is done so that the network learns from options with different expiries together instead of only one expiry. With this, we saw that during calibration, the optimizer compares Heston implied prices with PAN smoothed target prices all included maturities at once. Additionally, the CCN is adapted to multi-maturity data by using inputs such as Heston price, strike, maturity, and risk-free rate to learn a final correction.

The parameters used in the Heston model can be compared across the 3, 6, and 12 month maturities to study how market expectations change over time. In particular, short-maturity parameters tend to capture immediate market uncertainty, while longer maturities exhibit more stable behavior, suggesting expectations of volatility normalization over time. The overall term structure indicates that the market prices higher uncertainty in the short run, with smoother dynamics at longer horizons. These results are largely consistent with financial intuition, although some variation may arise from calibration sensitivity, data noise, and the use of smoothed inputs through the PAN framework.

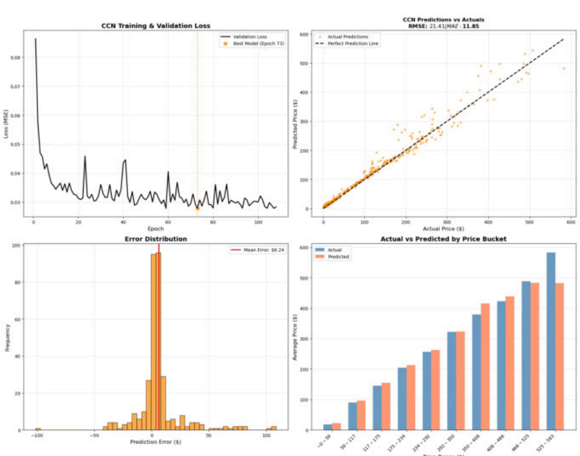
This model does have some strengths and weaknesses. Ultimately there are always tradeoffs when using these types of models that need to be balanced and closely looked into. A major benefit of using this model is the fact that the extended framework uses multiple maturities while retaining fast computation thanks to the surrogate pricing network. This makes it more realistic than a single maturity calibration.

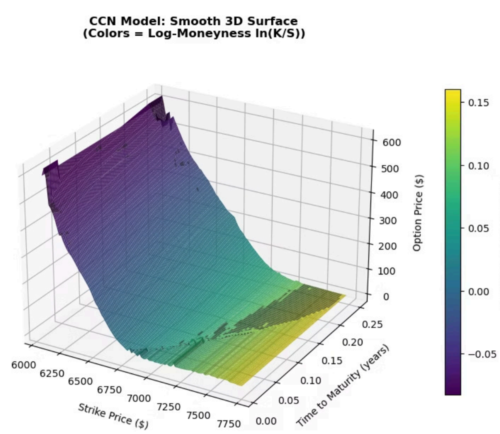
The drawbacks include the fact that calibration may still be sensitive to the initial guess, optimization settings, and parameter bounds. Different parameter combinations may produce similar prices which can make it hard to uniquely identify some parameters while trying to see how they change things around. Moreover, noisy prices and sparse strikes can affect results. Lastly, surrogate and correction networks improve speed and fit, but also add another layer of approximation on top of the original Heston model.

Overall, this framework is powerful, however the results should be interpreted with awareness of optimization sensitivity, data quality, and approximation error. Extending the framework to multiple maturities makes the model more useful because it captures variation across term structure rather than focusing on single expiry. The code includes PAN/CCN comparison plots based on RMSE, MAE, and MRE, showing that model accuracy is a central output of the framework.

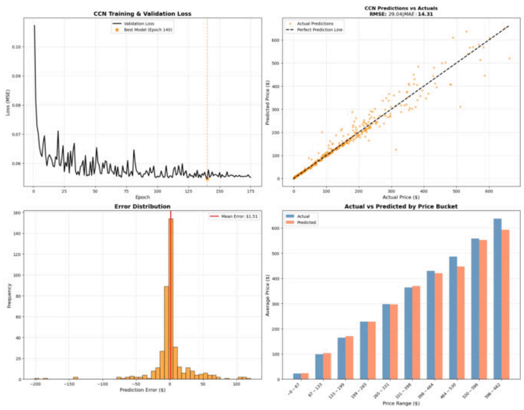
Figure 6: Traditional Model vs Heston

This calibrated figure illustrates the performance of the traditional Heston stochastic volatility model when calibrated to observed market option prices across different strike prices. The model successfully captures the overall downward relationship between strike price and option value and provides a reasonable fit near the at-the-money region around the spot price. There are however noticeable deviations remain at extreme strike levels, particularly for deep in-the-money and far out-of-the-money options where the model tends to overprice or underprice relative to market observations.





The above figure denotes 3-month maturity produced a Real RMSE of $21.41, Real MAE of $11.85, and Real MRE of 37.53%. Although the model captures the overall pricing pattern, the percentage error is highest among the three maturities. This suggests that short-dated options are harder to model accurately due to their greater sensitivity to immediate market volatility and short-term uncertainty. Despite this, the calibration still provides a reasonable fit and captures the main market pricing structure.



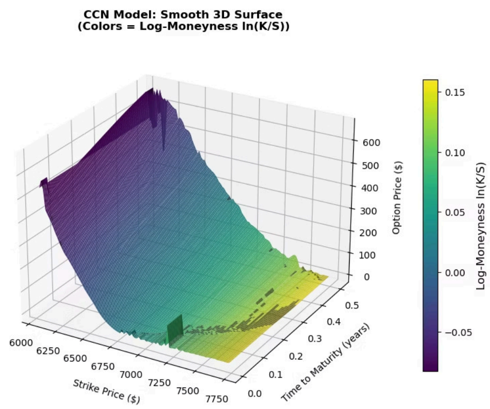
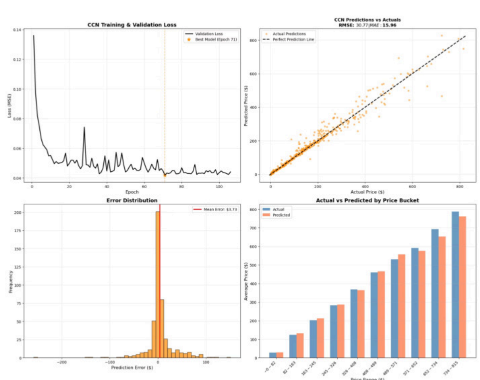


Figure 7: Multi-Maturity Calibration Results for the 6-Month Horizon

The above figure now denotes the 6 month calibration. Here we can see that the CCN framework produces a Real RMSE of $29.04,Real MAE of $14.31, and Real MRE of 16.50%. Although the absolute error measures of RMSE and MAE are higher than the 3-month maturity, the percentage error is significantly lower. This suggests that the model achieves a stronger relative fit for medium-term options because 6-month contracts are less sensitive to short-term market shocks while still retaining meaningful pricing structure. The additional time to maturity smooths some of the near-term volatility noise, making calibration more stable. These results indicate that the 6 month horizon may provide the best balance between market realism and calibration tractability within the multi-maturity framework.



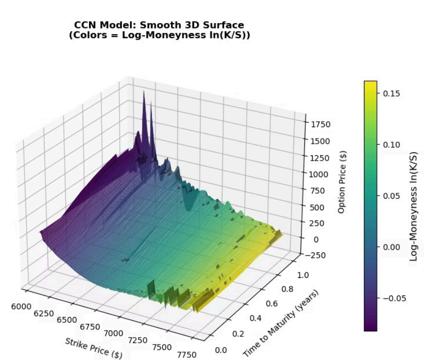


Figure 8: Multi-Maturity Calibration Results for the 6-Month Horizon

The 12-month maturity produced a Real RMSE of $30.77, Real MAE of $19.49, and Real MRE of 19.11%. Although the absolute dollar errors are the largest of the three maturities, the percentage error remains relatively low. This suggests that the model still performs well for long-dated options, which are generally less sensitive to short-term volatility shocks but involve larger contract values. Overall, the framework maintains a strong fit at the 12-month horizon.

Interpretation of Parameters Across Maturities

The calibrated Heston parameters become most informative when compared across multiple maturities, since they provide insight into how the market prices uncertainty over different time horizons. Rather than viewing the parameters only as numerical outputs, examining their movement across the 3, 6, and 12 month calibrations allows us to interpret how expectations of volatility, persistence, and downside risk evolve through time. In this sense, the parameter term structure transforms the quantitative calibration results into financially meaningful conclusions.

The mean reversion parameter κ reaches its highest value at the 6-month maturity, suggesting that the market expects volatility shocks to normalize most strongly over the medium-term horizon. This implies that while short-term uncertainty remains elevated, investors may anticipate stabilization within several months. At the 12-month maturity, the lower value κ may indicate longer term uncertainty becomes more structural.

The long run variance θ declines as maturity increases, producing a downward slope volatility structure. This suggests that current market conditions contain elevated near-term uncertainty. However volatility is expected to moderate more gradually over longer horizons. This is common when markets are reacting to macroeconomic stress.

The volatility-of-volatility parameter σ displays a U-shaped pattern across maturities. It is relatively high at the 3-month horizon, falls at 6 months, and then rises again at 12 months. This indicates that the market expects variance to stabilize in the medium term, before uncertainty returns at longer horizons due to unresolved long-term risks such as economic growth, interest rates, or policy conditions.

The correlation parameter ρ remains strongly negative across all maturities, which is financially intuitive for equity markets. This negative relationship reflects the leverage effect in which declining asset prices are associated with rising volatility. The persistence of negative correlation across maturities also helps explain the steep implied volatility skew commonly observed in index options.

The initial variance parameter v_0 remains close to its lower boundary and changes slightly across maturities suggesting the short-term calibration may have encountered numerical constraints, causing the optimizer to push the estimate toward a boundary value. As identified in the calibration results, this issue was later improved through the CCN which refined the final pricing surface and reduced residual errors.

Overall, the parameter evolution suggests a market environment characterized by elevated short-term volatility, relative stabilization around the 6-month horizon, and renewed uncertainty over the longer term. The consistently negative correlation indicates persistent downside risk throughout the maturity spectrum. These findings demonstrate that the multi-maturity framework provides economically meaningful insight beyond pricing accuracy alone, allowing the calibrated Heston model to reflect changing market expectations across time.

Limitations and Challenges

Though the Heston Model is a practical tool and used widely across the market, there are limitations and drawbacks which must be acknowledged. Firstly, the Heston model relies on simplifying assumptions that may not fully capture real market behavior. Particularly, the model assumes continuous price movements and does not include sudden jumps caused by earnings announcements, macroeconomic news, or market shocks. This can cause the model to struggle during periods of extreme volatility or rapid price dislocations. Additionally, Heston parameters are typically treated as constant during calibration in which real markets often experience changing volatility regimes over time. This can reduce pricing accuracy when market conditions shift quickly.

A second challenge concerns the calibration process. The optimization routine searches for parameters that minimize pricing error, however can be sensitive to initial parameter guesses, bounds, and numerical settings. Different combinations of parameters may generate similar option prices, making the calibration problem non-unique. Consequently, small changes in the data or optimization procedure may lead to different calibrated parameter values. This issue becomes more relevant when calibrating across many strikes and maturities simultaneously, since the model must balance fit quality across a broader range of contracts.

Quality of input data plays an important role. Market option data could have illiquid contracts, stale quotes, or missing observations. Although filtering procedures were used to improve data reliability, removing too many observations will reduce sample size and potentially bias the calibration toward liquid contracts. The final results remain dependent on the availability and cleanliness of the market data used.

The PAN, CCN, and surrogate network significantly improve computational speed and predictive accuracy while also introducing additional approximation layers. This is seen with smoothing raw market prices which helps create a more stable optimization surface, yet may also remove subtle market information. Similarly, surrogate neural networks approximate the Heston pricing map rather than solving it directly, meaning some model error may remain. The full framework therefore trades some interpretability for gains in speed and flexibility.

Finally, the multi-maturity extension provides a more realistic calibration environment but also introduces complexity. Using one parameter set to explain options across several maturities may oversimplify the true term structure of volatility due to the fact that market expectations can differ substantially between short-dated and long-dated contracts. Future improvements could include jump-diffusion extensions, time-varying parameters, regime-switching volatility models, and more robust optimization techniques. Despite these limitations, the overall framework still provides a valuable bridge between classical stochastic volatility modeling and modern deep learning methods.

CONCLUSION

This project began with a focus in quantitative finance. We postulated that the Heston stochastic volatility model provides a more realistic framework than Black-Scholes since its difficult and computationally expensive to calibrate in practice. Traditional calibration methods require repeated pricing evaluations struggle with noisy market data which makes them slow and unstable for real world implementation. The objective of our project was to develop a more practical framework that preserves the financial strengths of the Heston model while improving calibration efficiency and pricing accuracy.

We introduced a hybrid deep learning framework consisting of synthetic data generation, a surrogate pricing network, the Price Approximator Network (PAN), and the Calibration Correction Network (CCN). These are the four main components of our model. The surrogate network was trained to approximate Heston prices rapidly, replacing expensive repeated numerical evaluations during optimization. With training, the PAN learned a smooth pricing surface from observed market prices, helping reduce noise and stabilize the initial calibration process. Finally, the CCN learned the remaining pricing errors after calibration and refined the model output to better align with observed market prices. Together, these components created a practical pipeline that combines financial modeling with modern machine learning methods.

The code we constructed includes PAN/CCN comparison plots based on RMSE, MAE, and MRE, showing that model accuracy is a central output of the framework. Figure 6 illustrates the performance of the traditional Heston stochastic volatility model when calibrated to observed market option prices across different strike prices. The model successfully captures the overall downward relationship between strike price and option value and provides a reasonable fit near the at-the-money region around the spot price. There are however noticeable deviations remain at extreme strike levels, particularly for deep in-the-money and far out-of-the-money options where the model tends to overprice or underprice relative to market observations. This highlights the practical limitations of the standard Heston framework mainly emphasizing constant parameter structure and difficulty capturing more complex market effects in the tails.

The model successfully captures the overall downward relationship between strike price and option value and provides a reasonable fit near the at-the-money region around the spot price. There are however noticeable deviations remain at extreme strike levels, particularly for deep in-the-money and far out-of-the-money options where the model tends to overprice or underprice relative to market observations. This highlights the practical limitations of the standard Heston framework mainly emphasizing constant-parameter structure and difficulty capturing more complex market effects in the tails.

Further extensions to this project could include jump diffusion or regime switching volatility models, time-varying parameters, more advanced neural network architectures, and larger datasets. We could also compare this framework directly against industry benchmark calibration methods in live market settings.

Bibliography

- A. Zadgar, S. Fallah, F. Mehrdoust, Deep Learning-Enhanced Calibration of the Heston Model: A Unified Framework, ArXiv (2025)
- M. Forde, E. Jacquier, A tutorial on the Heston model, Quantitative Finance 11 (11) (2011) 1613–1624.
- S. Gu, B. Kelly, D. Xiu, Empirical asset pricing via machine learning, The Review of Financial Studies 33 (5) (2020) 2223–2273.
- J. B. Heaton, N. G. Polson, J. H. Witte, Deep learning in finance, Annual Review of Financial Economics 9 (2017) 145–165.
- S. Chen, P. Glasserman, Estimating the option price surface with a regression-based approach, Mathematical Finance 20 (2) (2010) 267–296.
- S. Becker, P. Cheridito, A. Jentzen, Pricing and hedging financial derivatives using neural networks, SIAM Journal on Financial Mathematics 10 (2) (2019) 747–777.
- S. Liu, A. Borovykh, L. A. Grzelak, C. W. Oosterlee, A neural network based framework for financial model calibration, Journal of Mathematics in Industry 9 (1) (2019) 9
- A. Paszke, S. Gross, F. Massa, et al., PyTorch: an imperative style, high-performance deep learning library, Advances in Neural Information Processing Systems 32 (2019) 8024-8035.