PHYSICS-INFORMED NEURAL NETWORKS FOR OPTION PRICING: THEORY AND MARKET APPLICATIONS

JIASI WANG AND JIMMY WU

Abstract. This work presents a hybrid approach to European call option pricing that combines financial theory and data-driven learning using Physics-Informed Neural Networks (PINNs). Our model integrates the Black-Scholes partial differential equation as a soft constraint alongside empirical financial data from AAPL options traded between 2016 and 2020. The model is trained based on a composite loss function that minimizes both the discrepancy with observed market prices and the residual of the Black-Scholes PDE (calculated via automatic differentiation and utilizing market implied volatility). An adaptive weighting scheme is incorporated to dynamically balance the influence of the data-fitting and physics-enforcing loss terms during training. We trained this model using data from 2016 to 2018 and assessing generalization performance on unseen data on 2019. Results demonstrate that this hybrid PINN can learn complex pricing relationships from market data and Black-Scholes' result, and this model is doing better than Black-Scholes formula in predicting the future option prices.

Contents

1. Introduction

The valuation of financial derivatives, particularly European call options, is fundamental in quantitative finance. Traditional methods rely on analytical models such as the Black-Scholes equation, which assumes frictionless markets and constant volatility. While elegant, these models often fail to capture real-world phenomena such as volatility smiles and liquidity effects[4].

This work explores the use of Physics-Informed Neural Networks (PINNs) to bridge the gap between theory and empirical market behavior. PINNs embed differential equations into the training of neural networks, allowing them to learn functions that satisfy physical (or financial) laws. This study proposes a Physics-Informed Neural Network (PINN) that leverages the Black-Scholes PDE while simultaneously learning from historical option price data.

Key words and phrases. Linear algebra.

2. Mathematical Formulation

This work addresses the problem of pricing European call options within the framework of the Black-Scholes model. The price of a European call option, denoted by u, is a function of the underlying asset price S and time t. Under idealized assumptions (constant volatility, constant risk-free interest rate, and the asset following geometric Brownian motion), this price is governed by what so-called Black-Scholes PDE.

2.1. The Black-Scholes PDE.

$$\frac{\partial u}{\partial t} + rS\frac{\partial u}{\partial S} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 u}{\partial S^2} - ru = 0,$$

- $u(S, t)$: the price of the option
- S: the underlying asset price ($S \geq 0$)
- t: the current time ($0 \leq t \leq T$)
- T: the option's expiration time
- r: the risk-free interest rate
- K: The strike price (exercise price) of the option.
- r: The constant risk-free interest rate.
- σ: The constant volatility of the underlying asset's price returns.

This PDE is defined over the spatio-temporal domain $S \in [0, \infty)$ and $t \in [0, T]$. [3]

2.2. Boundary conditions.

- **Terminal Condition (at expiry, $t = T$):** The value of the option at maturity is its intrinsic value, which is the maximum of zero and the difference between the stock price and the strike price.

$$u(S, T) = \max(S - K, 0)$$

- **Boundary Condition (as stock price approaches zero, $S \to 0$):** If the stock price is zero, the call option will be worthless for any $t < T$.

$$u(0, t) = 0, \quad 0 \leq t < T$$

- **Boundary Condition (as stock price approaches infinity, $S \to \infty$):** As the stock price becomes very large, the call option behaves like the underlying stock itself (minus the discounted strike price). This can be expressed as the option's delta approaching one.

$$\lim_{S\to\infty} \frac{\partial u}{\partial S}(S, t) = 1, \quad 0 \leq t < T$$

2.3. **Parameters.** The classical Black-Scholes PDE assumes the parameters σ and r are constant. However, in real financial markets, option prices depend on varying strike prices (K) and time to maturities ($T-t$), and it also depend implicitly on parameters like implied volatility (σ) and interest rates (r) which are not actually constant across all options or over time. Therefore, we treat this as a parametric problem, where the price u is approximated as a function of the current stock price, time, strike price, and relevant market-implied parameters. Our hybrid PINN framework aims to learn a function $u(S, t, K, \sigma_{IV}, r)$ that both satisfies the Black-Scholes PDE form using the local parameters from market data and matches observed market prices.

3. PINN-Based PDE Solver

The core of our approach is a neural network framework designed to approximate the parametric solution of the Black-Scholes PDE, $u(S, t, K, \sigma, r)$, by combining information from observed market data and the physical laws described by the PDE itself.

3.1. **Neural Network Architecture.** We utilize a feedforward neural network, denoted as $u_\theta(S_n, t_n, K_n, \sigma_n)$, parameterized by θ, to approximate the option price. [1] The network takes four inputs representing the normalized underlying asset price (S_n), normalized time to maturity (t_n), normalized strike price (K_n), and normalized implied volatility (σ_n), derived from the market data. The network consists of an input layer, three hidden layers with a specified number (128) of neurons (defined by HIDDEN_DIM) using the hyperbolic tangent (tanh) activation function, and a single output neuron representing the predicted option price. To ensure physical feasibility, a Rectified Linear Unit (ReLU) is applied to the final output layer to constrain the predicted option prices to be non-negative.

3.2. **Hybrid Loss Function Formulation.** The neural network is trained by minimizing a **composite loss function** that incorporates both market observations and the Black-Scholes PDE constraint. This approach is essential as purely data-driven models may violate theoretical principles, while solely physics-informed models might not capture the nuances of real market pricing. The total loss L_{total} is a combination of a data loss (L_{data}) and a physics-informed PDE loss (L_{pde}).

- **Data Loss (L_{data}):** This term measures the discrepancy between the network's predicted option price $u_\theta(S_n, t_n, K_n, \sigma_n)$ for a given market data point and the actual observed market price C_{LAST}. We use the Mean Squared Error (MSE) as the metric for this loss:

$$L_{data} = \frac{1}{N}\sum_{i=1}^{N}\left(u_\theta(S_{n,i}, t_{n,i}, K_{n,i}, \sigma_{n,i}) - C_{LAST,i}\right)^2$$

  where N is the number of data points in the current batch.
- **PDE Loss (L_{pde}):** This term enforces the Black-Scholes PDE at the spatio-temporal and parametric locations provided by the market data points. For each data point $(S_i, t_i, K_i, \sigma_{IV,i}, C_{LAST,i})$, we evaluate how well the network's output u_θ satisfies the PDE. This requires computing the necessary partial derivatives of u_θ with respect to S and t, specifically $\frac{\partial u_\theta}{\partial t}$, $\frac{\partial u_\theta}{\partial S}$, and $\frac{\partial^2 u_\theta}{\partial S^2}$.

  Using automatic differentiation provided by PyTorch, we compute the derivatives of the network's output with respect to its *normalized* inputs (S_n, t_n). We then apply the chain rule to transform these into derivatives with respect to the *unnormalized* physical variables (S, t). For example, $\frac{\partial u}{\partial S} = \frac{\partial u}{\partial S_n} \cdot \frac{\partial S_n}{\partial S} = \frac{\partial u}{\partial S_n} \cdot \frac{1}{\sigma_S}$, where σ_S is the standard deviation of the training S data used for normalization.

  The PDE residual is calculated by plugging these derivatives, the unnormalized market stock price (S), the assumed risk-free rate (r_{fixed}), the network output (u_θ), and crucially, the market implied volatility (σ_{IV}) into the Black-Scholes PDE formula. The PDE loss is the Mean Squared Error of this residual over the data points in the batch:

$$L_{pde} = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{\partial u_\theta}{\partial t} + rS\frac{\partial u_\theta}{\partial S} + \frac{1}{2}\sigma_{IV}^2 S^2 \frac{\partial^2 u_\theta}{\partial S^2} - ru_\theta\right)_i^2$$

  where the derivatives and u_θ are evaluated at $(S_i, t_i, K_i, \sigma_{IV,i})$. Note that the volatility σ in the PDE is taken as the market implied volatility σ_{IV} from the data point, representing a common practice in financial modeling when theoretical process volatility is unknown.

3.3. **Adaptive Loss Weighting.** Instead of fixing the weights for L_{data} and L_{pde}, we employ an adaptive weighting scheme inspired by multi-task learning.[2]We introduce two learnable parameters, log_var_{data} and log_var_{pde}, representing the logarithm of the variances associated with each loss term. The total loss optimized by the network is defined as:

$$L_{total} = \exp(-log\_var_{data})L_{data} + \frac{1}{2}log\_var_{data} + \exp(-log\_var_{pde})L_{pde} + \frac{1}{2}log\_var_{pde}$$

This formula come from maximum likelihood estimation. The terms $\exp(-log\_var)$ act as dynamic weights (inversely proportional to learned variance), while the $\frac{1}{2}log\_var$ terms serve as regularizer. By optimizing log_var_{data} and log_var_{pde} alongside the network parameters via gradient descent, the model learns to balance the influence of the data and PDE constraints automatically based on their relative scales and noise levels during training.

3.4. **Training Procedure.** The network parameters θ, log_var_{data}, and log_var_{pde} are optimized using the Adam optimizer. We use a DataLoader to process the large dataset in mini-batches. Within each training step, the data loss and PDE loss are calculated for the current batch. The total adaptive loss is computed, and gradients are calculated via backpropagation. The optimizer then updates all parameters. We utilize Automatic Mixed Precision (AMP) with a GradScaler to accelerate training on compatible GPUs by performing certain operations in lower precision while maintaining numerical stability. A learning rate scheduler (*ReduceLROnPlateau*) is used to adjust the learning rate during training if the total loss plateaus.

4. Market Data Modeling

The proposed framework operates as a hybrid model, directly utilizing a large dataset of real-world financial market observations alongside the theoretical constraints of the Black-Scholes PDE. This section details the preparation and integration of the market data into the PINN training process.

4.1. **Data Preprocessing.** We use a historical dataset of AAPL European call option from 2016 to 2020, obtained from a CSV file. Each record includes relevant information such as the quote date, expiration date, underlying asset price, strike price, implied volatility, and the option's last traded price.

The initial data undergoes a crucial preprocessing pipeline:

- **Column Cleaning:** Raw column headers are cleaned to remove extraneous characters and whitespace, ensuring accurate data access.
- **Column Selection:** Only the essential columns required for option pricing (QUOTE_DATE, EXPIRE_DATE, UNDERLYING_LAST, C_LAST, STRIKE, C_IV) are selected.
- **Date and Time Calculation:** Quote dates and expiration dates are parsed into datetime objects. The time to maturity $(T - t)$ is calculated in years based on the difference between these dates.
- **Numeric Conversion:** Relevant columns are converted to numeric data types.
- **Data Filtering:** Rows with missing values in critical fields or invalid conditions (e.g., non-positive time to maturity, zero prices, zero strikes, zero underlying prices, zero implied volatility) are removed to ensure data quality and numerical stability.

4.2. **Temporal Data Splitting.** To evaluate the model's ability to generalize to unseen market conditions, the dataset is split based on time. Data with a quote date up to and including 2018 is designated as the **training set**, while data with a quote date after 2018 (i.e., 2019) is reserved as the **test set**. This temporal split is critical for assessing predictive performance on future market dynamics that were not present during training.

4.3. **Data Normalization.** To improve the training stability and convergence speed of the neural network, the input features (S, t_{years}, K, σ_{IV}) are normalized. We apply standardization (z-score normalization):

$$x_n = \frac{x - \mu_x}{\sigma_x} \tag{1}$$

where x is the original feature value, x_n is the normalized value, and μ_x and σ_x are the mean and standard deviation of the feature, respectively.

The mean and standard deviation for each feature $(\mu_S, \sigma_S, \mu_t, \sigma_t, \dots)$ are calculated solely from the training set. These normalization constants are then applied consistently to both the training set and the test set to prevent data leakage from the future into the training process. These constants are saved alongside the trained model for future inference on new data. The option price (C_{LAST}) remains unnormalized since it is the target output.

4.4. **Data Integration into the PINN Loss.** The processed and normalized market data are integrated into the PINN training through the composite loss function (Section 3.2).

- L_{data}: For each batch of data, the network's prediction is directly compared to the observed C_{LAST} value, contributing to L_{data}. This term anchors the model to the empirical price surface.
- L_{pde}: The same input features from the data batch (S, t_{years}, K, σ_{IV}) are used as the evaluation points for the Black-Scholes PDE residual. L_{pde} is calculated by assessing how well the network's predicted price and its derivatives (at these data-defined locations and parameters) satisfy the PDE. The market's implied volatility (σ_{IV}) for each option is explicitly used as the volatility parameter (σ) in the PDE evaluation for that data point. The risk-free rate (r_{fixed}) is held constant across all evaluations as a simplification.

The training process uses DataLoaders to efficiently feed batches of the normalized training data (features and target price) to the model, calculating both loss components in each step and updating the model parameters and adaptive weights via gradient descent.

5. Results and Evaluation

This section presents the results obtained from training the hybrid PINN model on the AAPL option data (2016-2018) and evaluating its performance on both the training set and the unseen test set from 2019.

5.1. **Training Process.** The model was trained for 1200 epochs using the Adam optimizer with adaptive loss weighting and Automatic Mixed Precision. The evolution of the unweighted data loss (L_{data}), unweighted PDE loss (L_{pde}), and the total adaptive loss (L_{total}) are shown in Figure 1. The figure also plots the learned log-variances $(log\_var_{data}$ and $log\_var_{pde})$ which determine the effective weights applied to L_{data} and L_{pde} during optimization.

As shown in Figure 1, the total loss decreased consistently throughout training, indicating successful optimization. The unweighted data loss also showed a significant downward trend, reflecting the network's ability to fit the historical market prices. The unweighted PDE loss started relatively low, increased initially as the network adjusted to fit the data, and then decreased at a low level towards the end of training. The learned log-variances adapted over time, indicating that the optimization process dynamically adjusted the effective weights for the data and PDE loss components.

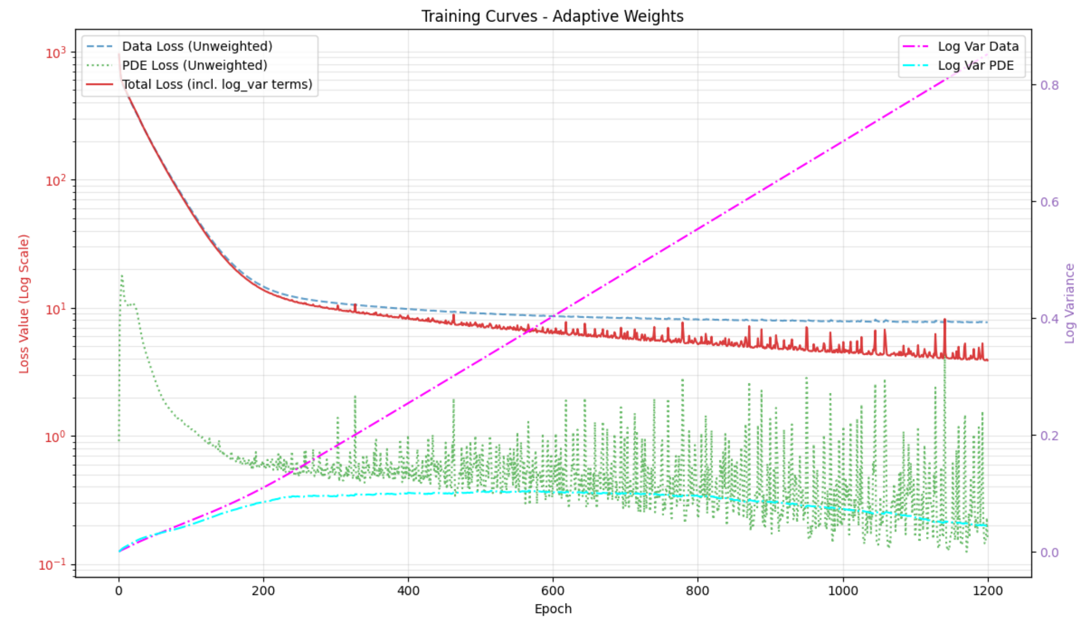

FIGURE 1. Training Curves: Evolution of unweighted Data Loss, unweighted PDE Loss, total adaptive Loss, and learned log-variances over epochs. Losses are shown on a logarithmic scale.

5.2. **Evaluation Metrics.** The performance of the trained hybrid PINN model was quantitatively evaluated using Mean Squared Error (MSE) and Mean Absolute Error (MAE) against the observed market prices (C_{LAST}) on the training set (2016-2018) and the test set from 2019. A Black-Scholes model using the market implied volatility (C_{IV}) as the volatility parameter and a fixed risk-free rate ($r_{fixed} = 0.05$) served as a baseline for comparison. The metrics are summarized in Table 1.

TABLE 1. MSE and MAE for PINN and Black-Scholes Baseline on Training and 2019 Test Sets.

Dataset	PINN MSE	PINN MAE	BS MSE	BS MAE
Training Set (2016-2018)	59.3276	3.8924	109.4576	4.0602
Test Set (Year 2019)	42.1923	3.5955	53.1854	4.0489

On the unseen 2019 test set, the hybrid PINN achieved a Mean Squared Error of 42.19 and a Mean Absolute Error of 3.60. Compared to the Black-Scholes baseline, which resulted in an MSE of 53.19 and an MAE of 4.05 on the same data, the hybrid PINN demonstrated superior accuracy. This indicates that the model, trained on prior data and physics constraints, successfully generalized to predict option prices in the subsequent year (2019) with lower error than the standard Black-Scholes formula using market implied volatility.

5.3. **Qualitative Analysis.** Visual inspection of the evaluation plots provides further insight into the models' performance on the 2019 test set.

Figure 2 shows scattered predictions and market prices over time for a subset of the 2019 data. Both PINN (red) and BS (blue) predictions generally track the market price fluctuations (green), though the PINN appears to show a slightly closer overall agreement as supported by the numerical metrics. The non-negative output constraint applied to the PINN is also consistently observed.

Figures 3 and 4 illustrate prediction errors as a function of strike price and moneyness (S/K), respectively, for a subset of the 2019 test data. These plots are crucial for identifying potential biases across different option types. The scatter plots show that the PINN errors (red) are generally smaller and potentially less systematically biased across strike prices and moneyness levels compared to the BS errors (blue), further supporting the quantitative findings for 2019.

The scatter plot of predicted price vs actual market price (Figure 5) for 2019 provides a direct visual correlation assessment. Points lying close to the $y = x$ line indicate accuracy. The concentration of PINN predictions (red) around the $y = x$ line appears visually tighter than the BS predictions (blue) for the 2019 data.

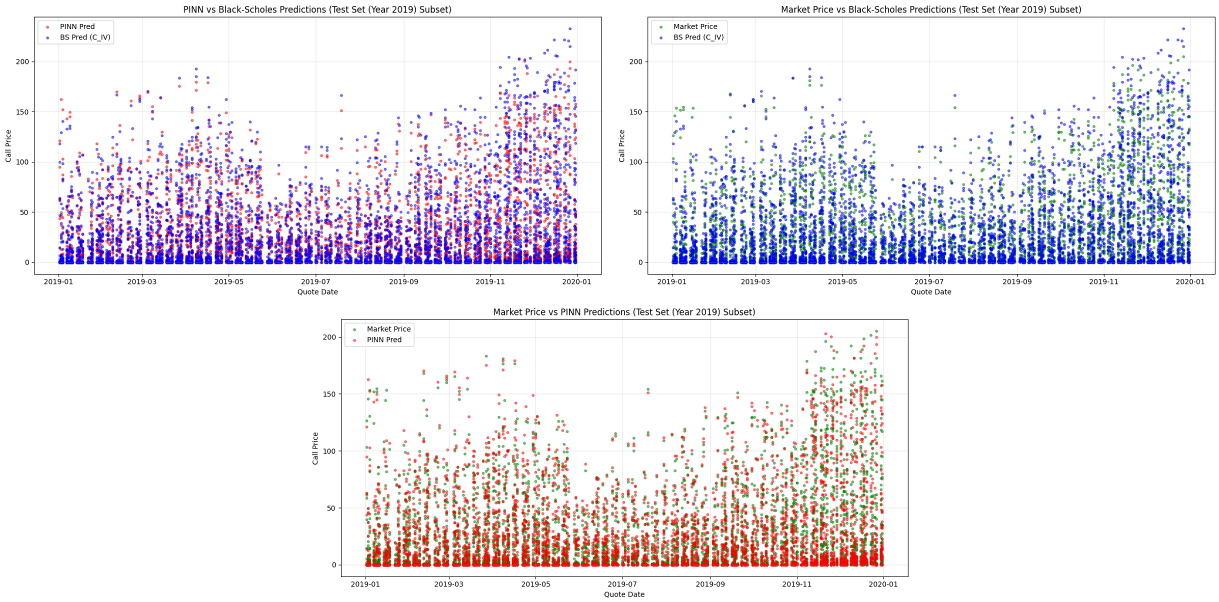

FIGURE 2. Time Series Comparison of PINN Predictions, Black-Scholes Predictions, and Market Prices for a subset of the 2019 Test Set.

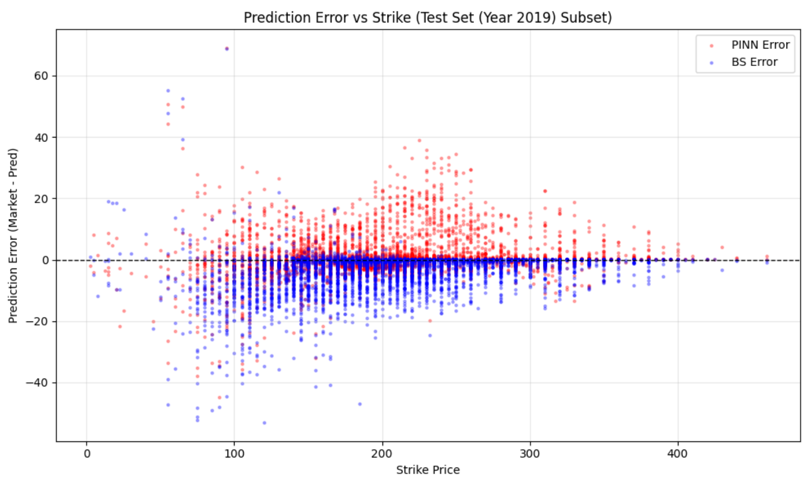

FIGURE 3. Prediction Error vs Strike Price for a subset of the 2019 Test Set.

Finally, the histogram of prediction errors (Figure 6) for the 2019 test set illustrates the distribution shape of errors for each model. The distribution for the PINN appears slightly narrower and potentially more centered around zero compared to the BS errors for this period.

6. Discussion

The results demonstrate the potential of the hybrid PINN framework for option pricing, while also highlighting key challenges when applying such models to real-world financial time-series data.

The superior performance of the hybrid PINN over the Black-Scholes baseline on the 2019 test set is a promising outcome. It suggests that the model that is trained on a combination of historical data and theoretical physics constraints, learned a pricing function that generalizes better to unseen data within a similar market regime compared to simply using market implied volatility in the standard BS formula. The physics-informed loss term likely acted as a valuable regularizer, encouraging a smoother, more theoretically consistent price surface than a purely data-driven model might achieve, thus aiding interpolation and short-term extrapolation into 2019.

However, the significant challenges that hybrid models face when encountering market conditions substantially different from the training data distribution must be acknowledged. Real financial markets are non-stationary; periods of stability can be punctuated by sudden shifts in volatility levels, correlations, and underlying price dynamics.

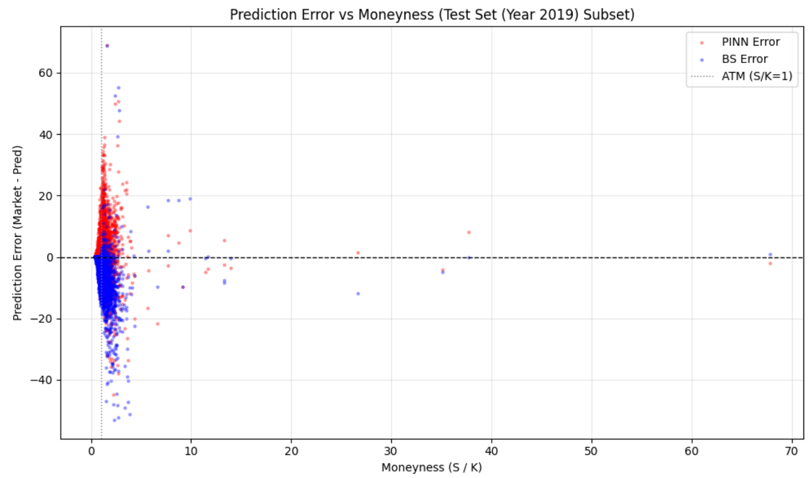

FIGURE 4. Prediction Error vs Moneyness (S/K) for a subset of the 2019 Test Set.

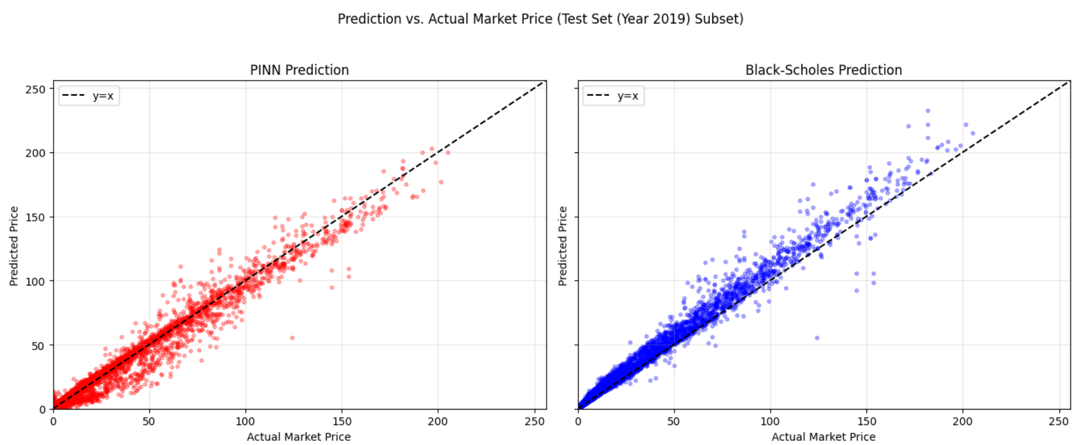

FIGURE 5. Predicted vs. Actual Market Price for the Hybrid PINN and Black-Scholes (BS) Baseline on a subset of the 2019 Test Set.

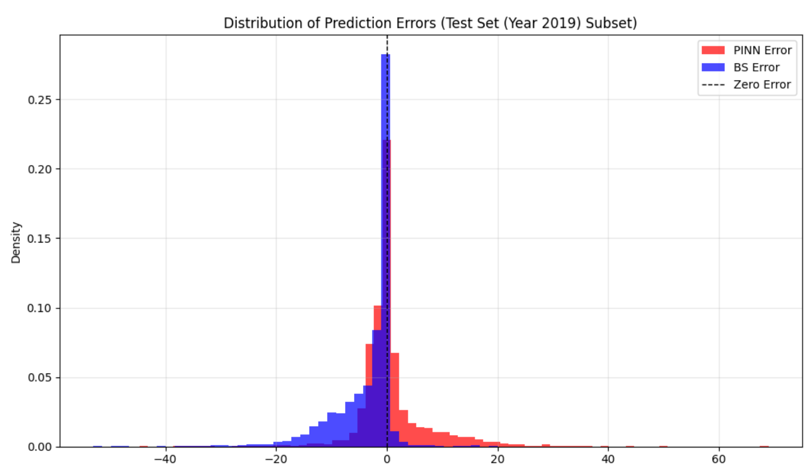

FIGURE 6. Distribution of Prediction Errors (Market Price - Predicted Price) for the Hybrid PINN and Black-Scholes (BS) Baseline on a subset of the 2019 Test Set.

A model trained solely on historical data may struggle to accurately predict prices when the underlying process significantly deviates from the learned distribution. Financial time-series analysis often reveals that models trained on past data are most effective for predicting the immediate future. The temporal split used here exposes the model to the natural evolution of market regimes, providing a more realistic test of generalization than random data splits.

The design choices in formulating the PDE loss also warrant discussion. Using the market implied volatility (σ_{IV}) directly within the Black-Scholes PDE residual calculation is a practical approach, aligning the physics check with the market's observed parameter for that option at that time. However, this differs from the theoretical assumption of σ being the constant volatility of the underlying process. Therefore, this model is checking against a PDE where the volatility parameter is non-constant across options and time, which is a deviation from the strict Black-Scholes PDE.

The adaptive loss weighting mechanism aimed to automatically balance the influence of the data and PDE terms. Analyzing the learned log-variances (Figure 1) provides insight into this process, showing how the optimization favored the data term initially before adjusting the relative weighting over time. The effectiveness of this adaptive balancing in finding the optimal trade-off in various market conditions remains an area for further investigation.

Limitations of the current model include the assumption of a fixed risk-free rate (r_{fixed}), which is not representative of real markets, and the reliance solely on the Black-Scholes PDE, which does not capture complex dynamics like stochastic volatility or jumps present in asset prices.

Future work could focus on incorporating dynamic interest rates, extending the framework to more advanced option pricing models (e.g. Heston model for stochastic volatility), or exploring different techniques for handling market regime shifts, such as transfer learning or continuous training on rolling windows of data. Further investigation into the interaction between the adaptive weighting scheme and market data characteristics is also warranted.

7. Conclusion

This project successfully develops and evaluates a hybrid physically-informed neural network framework for parametric option pricing that integrates real market data and the Black-Scholes PDE.The model demonstrates good generalization capabilities and outperforms the Black-Scholes baseline using market-implied volatility on the 2019 test set. This highlights the potential of combining physical constraints with empirical data to improve forecast accuracy under relevant market conditions. The framework leverages modern deep learning techniques, including physically-enforced auto-differentiation, adaptive loss weighting, batch processing, and blended accuracy, to enable efficient training on large datasets. While the model demonstrated effectiveness in the 2019 market regime, the challenges inherent in generalizing it to significantly different, highly volatile market periods highlight the complexity of financial time series modeling and suggest avenues for future research to build more robust, adaptive pricing models.

References

[1] Dhiman, Ashish, and Yibei Hu. "Physics Informed Neural Network for Option Pricing." *arXiv preprint* arXiv:2312.06711 (2023).

[2] Kathane, Shivprasad, and Shyamprasad Karagadde. "A Physics Informed Neural Network (PINN) Methodology for Coupled Moving Boundary PDEs." *arXiv preprint* arXiv:2409.10910 (2024).

[3] Black, Fischer, and Myron Scholes. "The Pricing of Options and Corporate Liabilities." *The Journal of Political Economy*, vol. 81, no. 3, May–June 1973, pp. 637–654. The University of Chicago Press. `http://www.jstor.org/stable/1831029`. Accessed 27 Sept. 2008.

[4] Murkute, Amod, and Tanuja Sarode. "Forecasting Market Price of Stock Using Artificial Neural Network." *International Journal of Computer Applications*, vol. 124, no. 12, 2015.

Carnegie Mellon University, Pittsburgh, PA 15213, USA
Email address, Author 1: `jiasiw@andrew.cmu.edu`

Carnegie Mellon University, Pittsburgh, PA 15213, USA
Email address, Author 2: `zhijingw@andrew.cmu.edu`