Advanced topics:
Semi-static replication using ANNs

Group 4: Divya Gajera (14932644) & Nitai Nijholt (12709018)

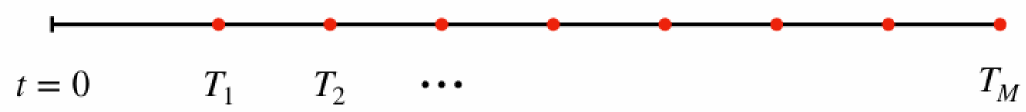
Objectives

- **Pricing**: Calculate the fair value of the derivative at any timepoint
- **Hedging**: Construct a semi-static hedge using short-maturity European options that replicate the derivative's payoff.
- **Estimate forward path sensitivities in complex derivatives**

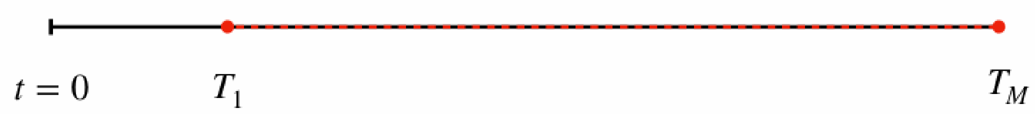
Bermudan Option

- European option:

- Bermudan option:



- American option:

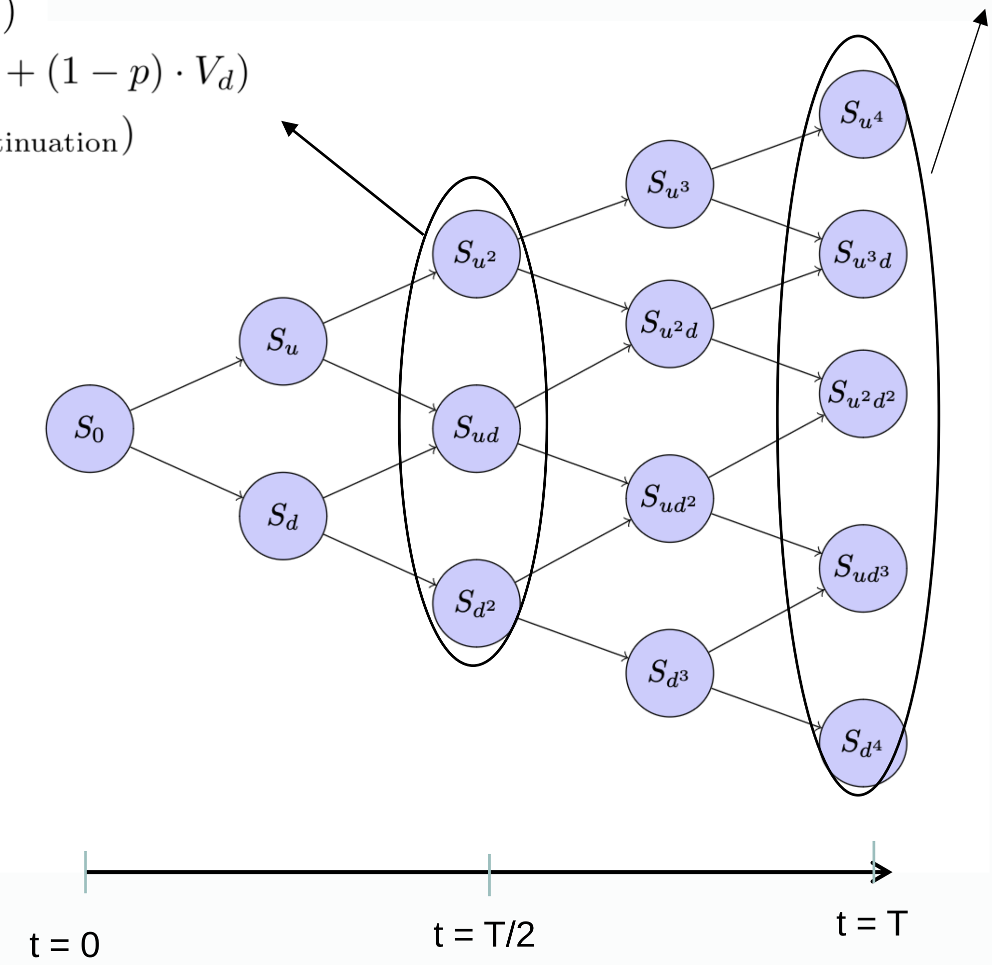


Binomial Method

$$V_{\text{intrinsic}} = \max(0, K - S_t)$$
$$V_{\text{continuation}} = e^{-r\Delta t}\left(p \cdot V_u + (1-p)\cdot V_d\right)$$
$$V_{\text{node}} = \max(V_{\text{intrinsic}}, V_{\text{continuation}})$$

$$V_T = \max(0, K - S_T)$$

$$\Delta = \frac{V_{t+1}^{\text{up}} - V_{t+1}^{\text{down}}}{S_{t+1}^{\text{up}} - S_{t+1}^{\text{down}}}$$



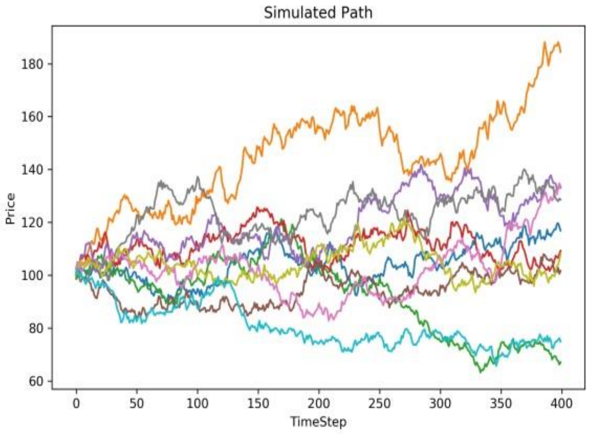
Monte Carlo Approach

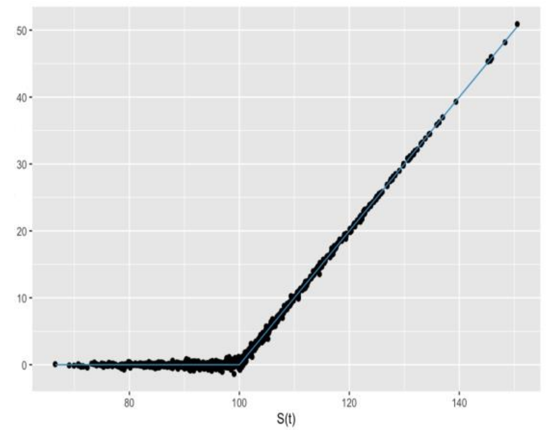
Monte Carlo Approach to approximate Bermudan option price consists of three main components:

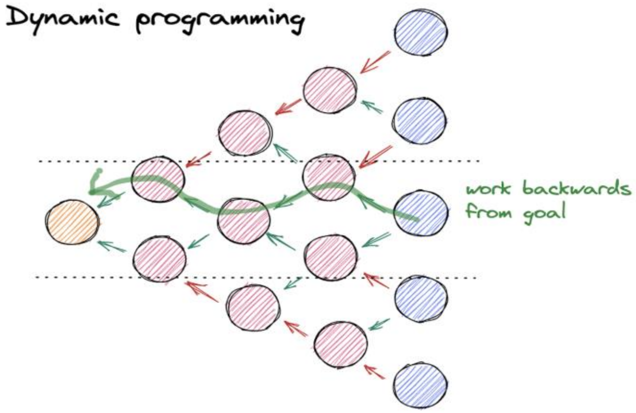
Simulations - Simulate a large number of price paths for the underlying asset using stochastic processes (e.g., Geometric Brownian Motion)

Regression - It fits the simulated future payoffs (from Monte Carlo paths) to the state variables (e.g., current asset prices).

Dynamic Programming - backward induction process that determines the optimal exercise strategy.







Monte Carlo Approach

The Monte Carlo methods for pricing Bermudan style options can be broadly divided in two categories :

1. **Regress Now** - Regression is performed at each exercise date using the current state variables (e.g., asset prices at that date) to approximate the continuation value. The continuation value at t_m is calculated directly based on the regression results.

2. **Regress Later** - Regression is deferred to later times, estimating the continuation value in terms of future states (e.g., payoffs at a future time step). The regression result at t_{m+1} is used retrospectively to decide the exercise policy at t_m

Motivation RLNN

Challenges in pricing complex derivatives

- Finite difference or Tree methods suffers from Curse of dimensionality

- Monte Carlo performs better, but high computational cost

- Dynamic hedging breaks down in low liquidity environment

Motivation RLNN

Advantages RLLN:

- (Semi) Static Hedging
- Interpretability
- Lower Computational Cost
- Universal Approximation Theorem

RLNN Algorithm

Main idea: Approximate pay-off of complex derivative using linear combination of simple European Options

Algorithm Steps:

1. Generate simulations of N underlying asset paths using GBM: $dS_t = \mu S_t \, dt + \sigma S_t \, dW_t$

2. Compute option payoffs at maturity for each path: $V_T^{(n)} = \max(h(S_T^{(n)}), 0)$

3. Train Neural Network $G(S)$ each time step to approx. option values: $\tilde{G}_{\beta_{t_m}}(S_{t_m}^{(n)}) \approx \tilde{V}_{t_m}^{(n)}$

4. Compute continuation value using the trained network: $Q_{t_{m-1}}^{(n)} = E\left[\tilde{G}_{\beta_{t_m}}(S_{t_m}) \mid S_{t_{m-1}}^{(n)}\right]$

5. Option Value Update: Update option values: $\tilde{V}_{t_{m-1}}^{(n)} = \max(h(S_{t_{m-1}}^{(n)}), Q_{t_{m-1}}^{(n)})$

6. Repeat step 3-5 backward to the initial time. $m = M-1, M-2, \ldots, 1,$ moving backward in time.

Image credit: Hoencamp, J. (2024, Fall). *Advanced topics: Semi-static replication using ANNs*. Advanced Topics in Computational Finance, University of Amsterdam.

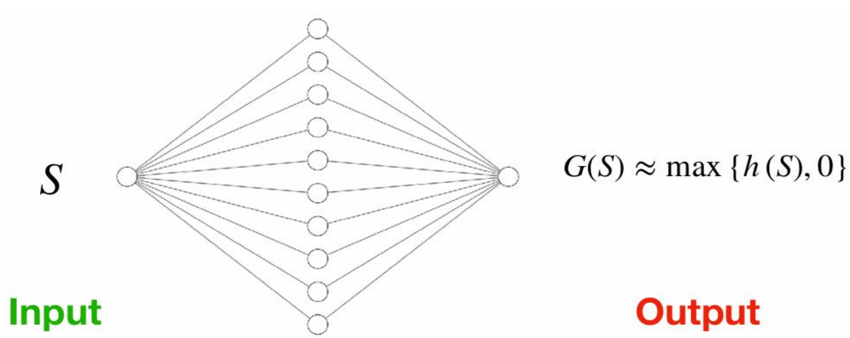
RLNN Algorithm

Main idea: Approximate pay-off of complex derivative using linear combination of simple European Options

NN structure:

- Hidden Layer Uses ReLU activation functions
- Each neuron represents payoff of single European option
- Portfolio weights correspond to w_i ; biases b_i to strike

Regress later because it uses feed forward neural network for the regression at each time step



$$\tilde{G}^{\beta_{t_m}}(\mathbf{x}) = \sum_{i=1}^{p} \omega_i \varphi\left(\mathbf{w}_i^{\top}\mathbf{x} + b_i\right) + b_2$$

Image credit: Hoencamp, J. (2024, Fall). *Advanced topics: Semi-static replication using ANNs*. Advanced Topics in Computational Finance, University of Amsterdam.

Expected value

$$Q_{t_{m-1}}^{(n)} = E\left[\tilde{G}_{\beta_{t_m}}(S_{t_m}) \mid S_{t_{m-1}}^{(n)}\right] \rightarrow \mathbb{E}\left[\varphi\left(\mathbf{w}_i^\top \mathbf{S}_{t_m} + b_i\right) \mid \mathbf{S}_{t_{m-1}}\right] \rightarrow \mathbb{E}[\max(w_i S_{t_m} + b_i, 0) | S_t]$$

Cases

Case 1: $w_i \geq 0$ and $b_i \geq 0$

The expected value simplifies to the price of a forward contract:

$$\text{Expected Value} = w_i \cdot S_{t_{m-1}} \cdot e^{r \cdot \Delta t} + b_i$$

Case 2: $w_i > 0$ and $b_i < 0$

This case represents a European call option. The strike price is:

$$\text{Strike} = -\frac{b_i}{w_i}$$

$$\text{Expected Value} = w_i \cdot \text{Black-Scholes}(S_{t_{m-1}}, \text{Strike}, r, \sigma, \Delta t, \text{option\_type='call'})$$

Case 3: $w_i < 0$ and $b_i > 0$

This case represents a European put option. The strike price is:

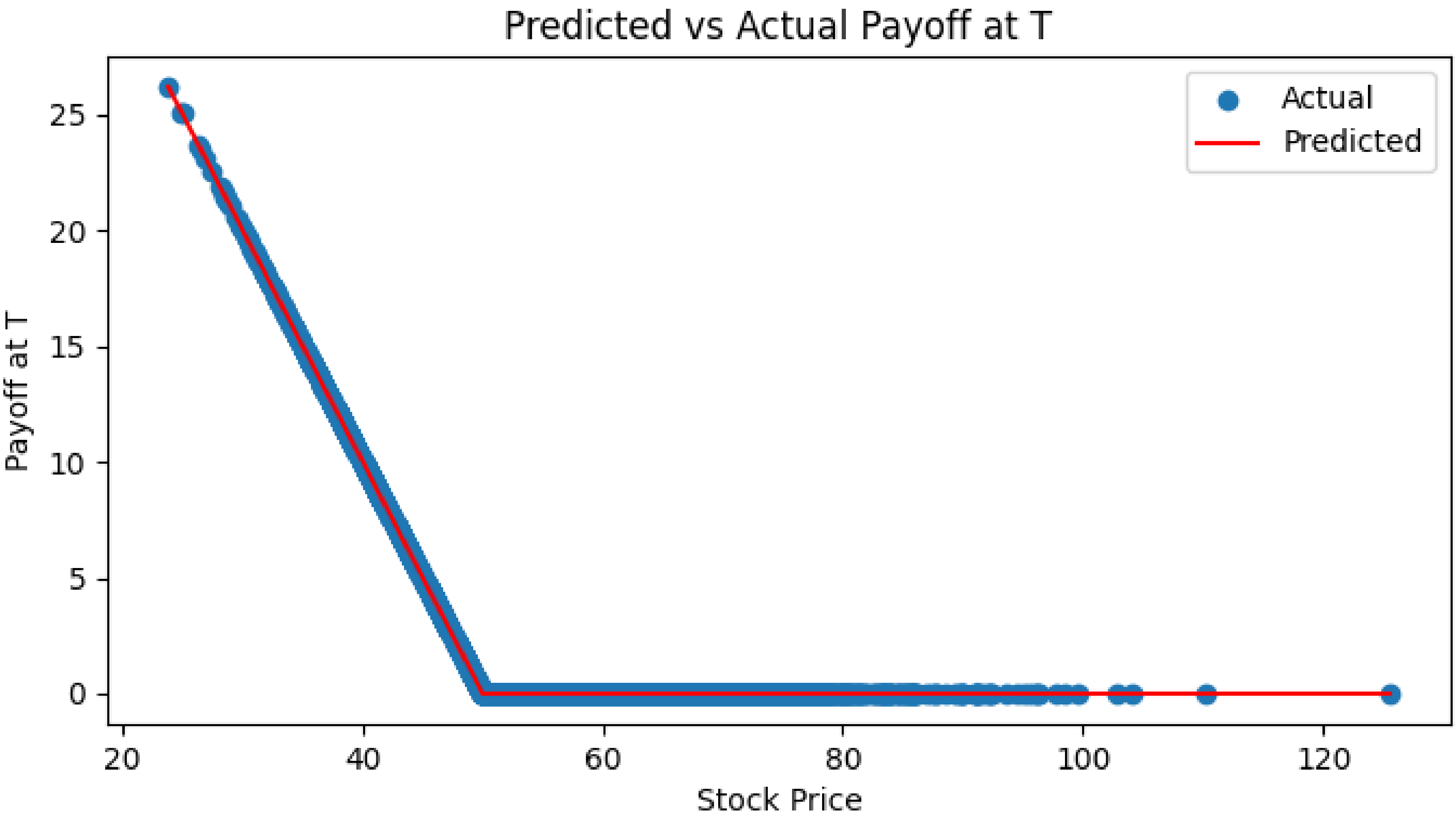
$$\text{Strike} = -\frac{b_i}{w_i}$$

$$\text{Expected Value} = -w_i \cdot \text{Black-Scholes}(S_{t_{m-1}}, \text{Strike}, r, \sigma, \Delta t, \text{option\_type='put'})$$

Case 4: $w_i \leq 0$ and $b_i \leq 0$

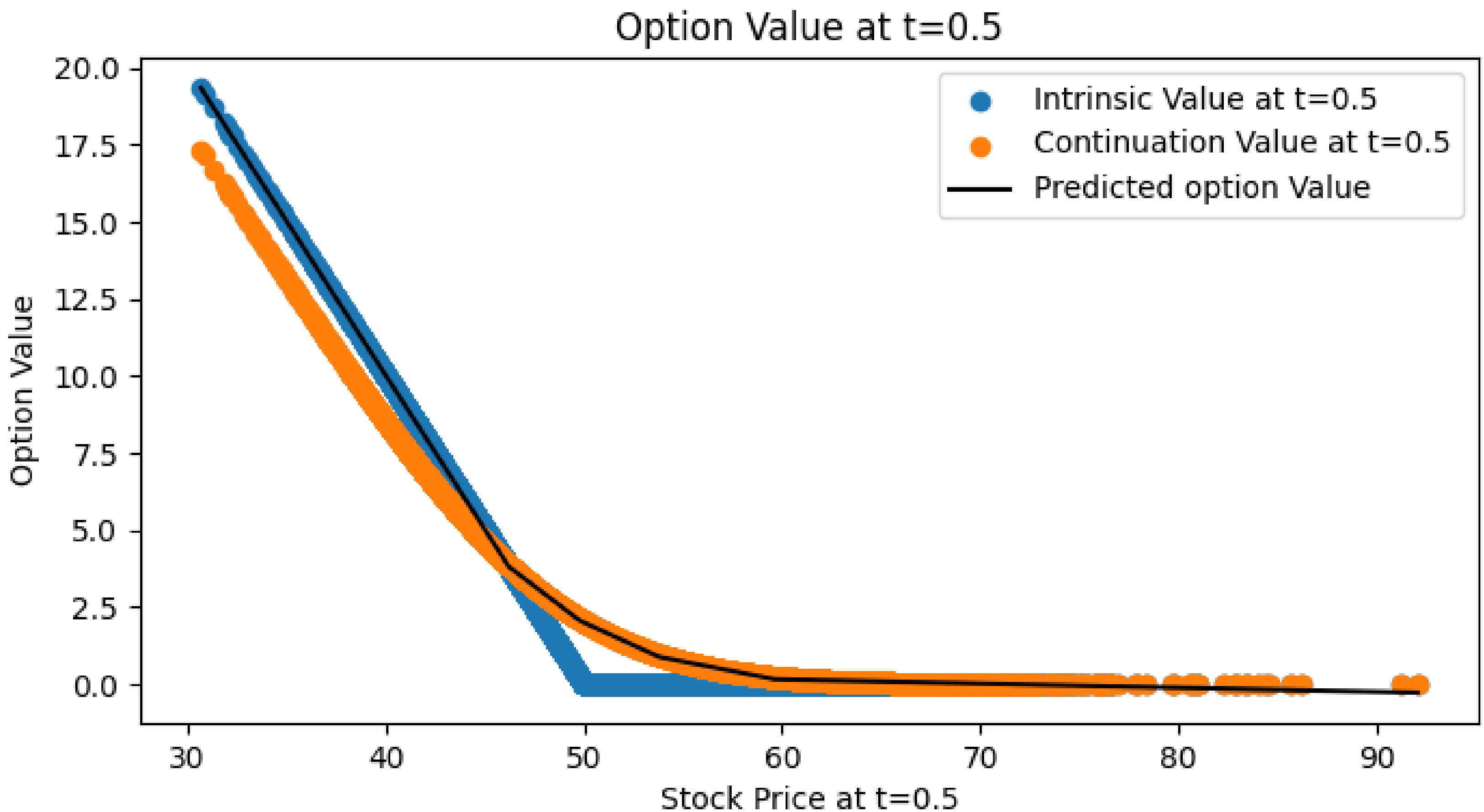
$$\text{Expected Value} = 0$$

Initial Results



Parameters: S0=50, K=50, μ=0.06, σ=0.2, T=1, N=4, M=10000

Initial Results

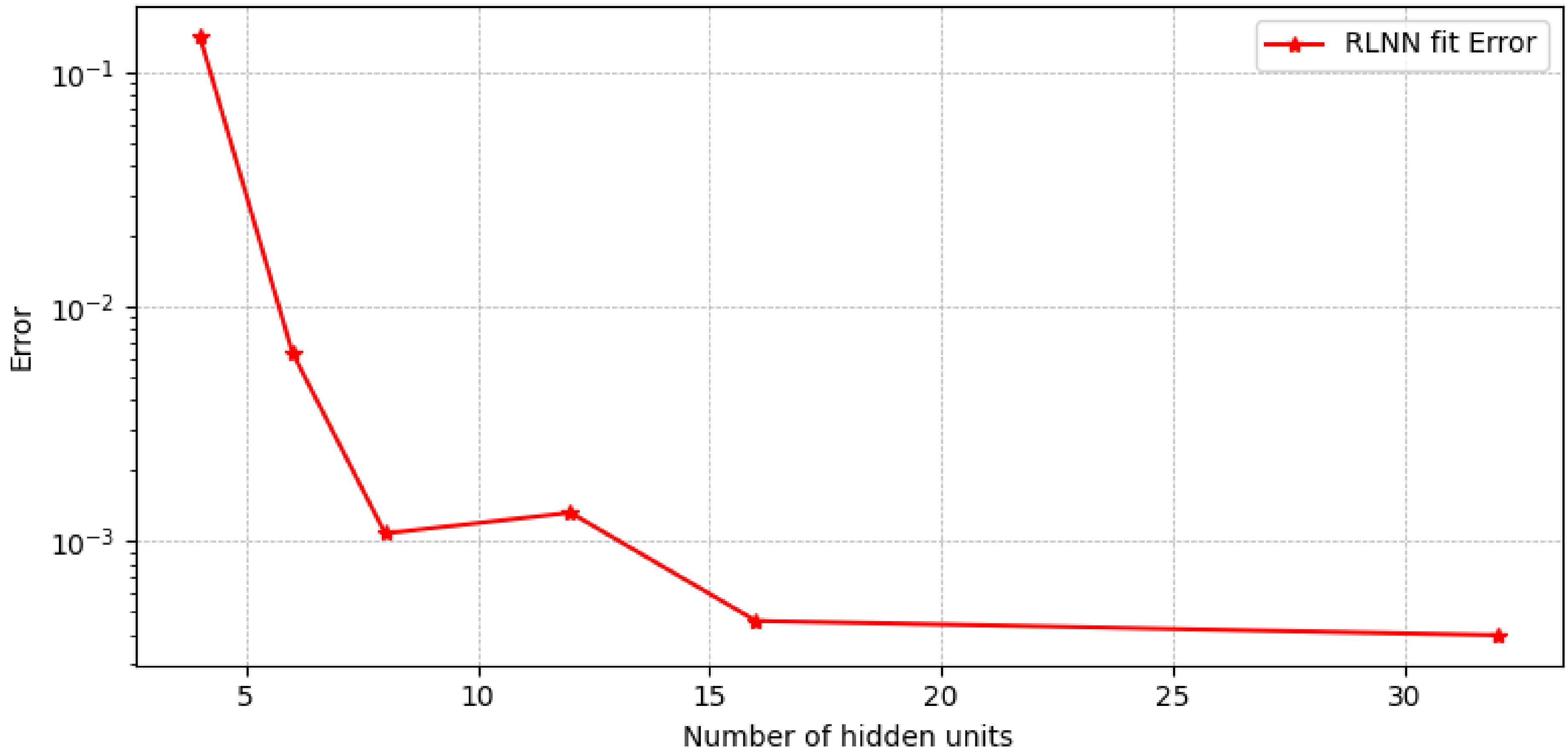


Parameters: S0=50, K=50, μ=0.06, σ=0.2, T=1, N=4, M=10000

Hyperparameter tuning

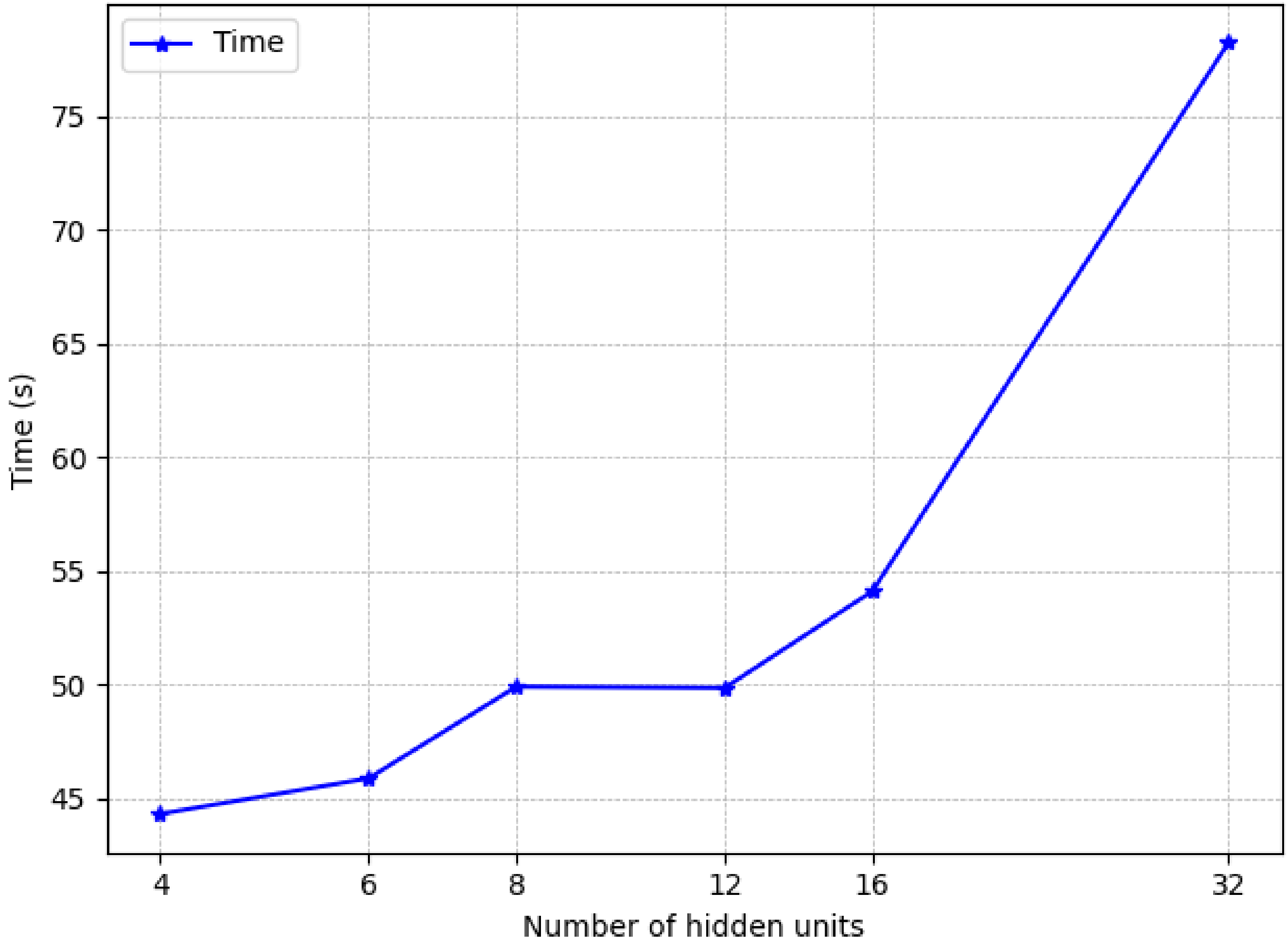
Parameters: S0=50, K=50, μ=0.06, σ=0.2, T=1, N=4, M=10000

Hyperparameter tuning



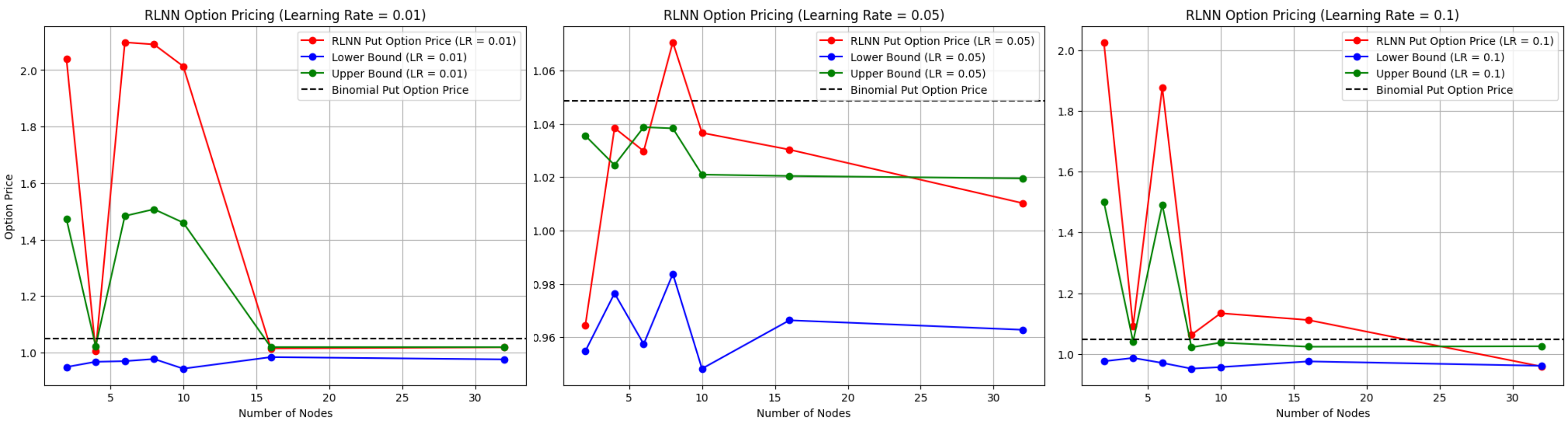
Parameters: S0=50, K=50, μ=0.06, σ=0.2, T=1, N=4, M=10000

Hyperparameter tuning



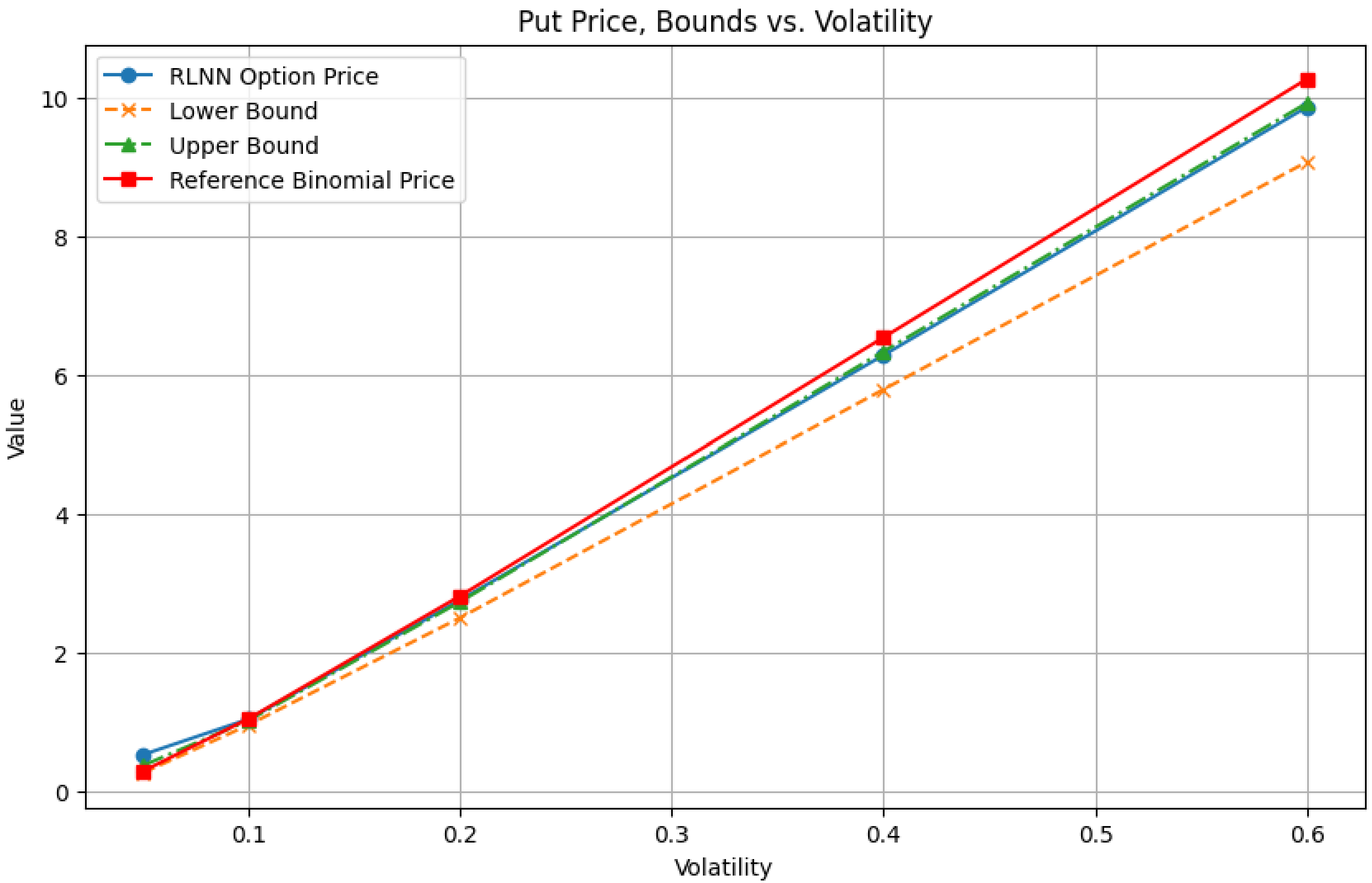
Parameters: S0=50, K=50, μ=0.06, σ=0.2, T=1, N=4, M=10000

Hyperparameter tuning



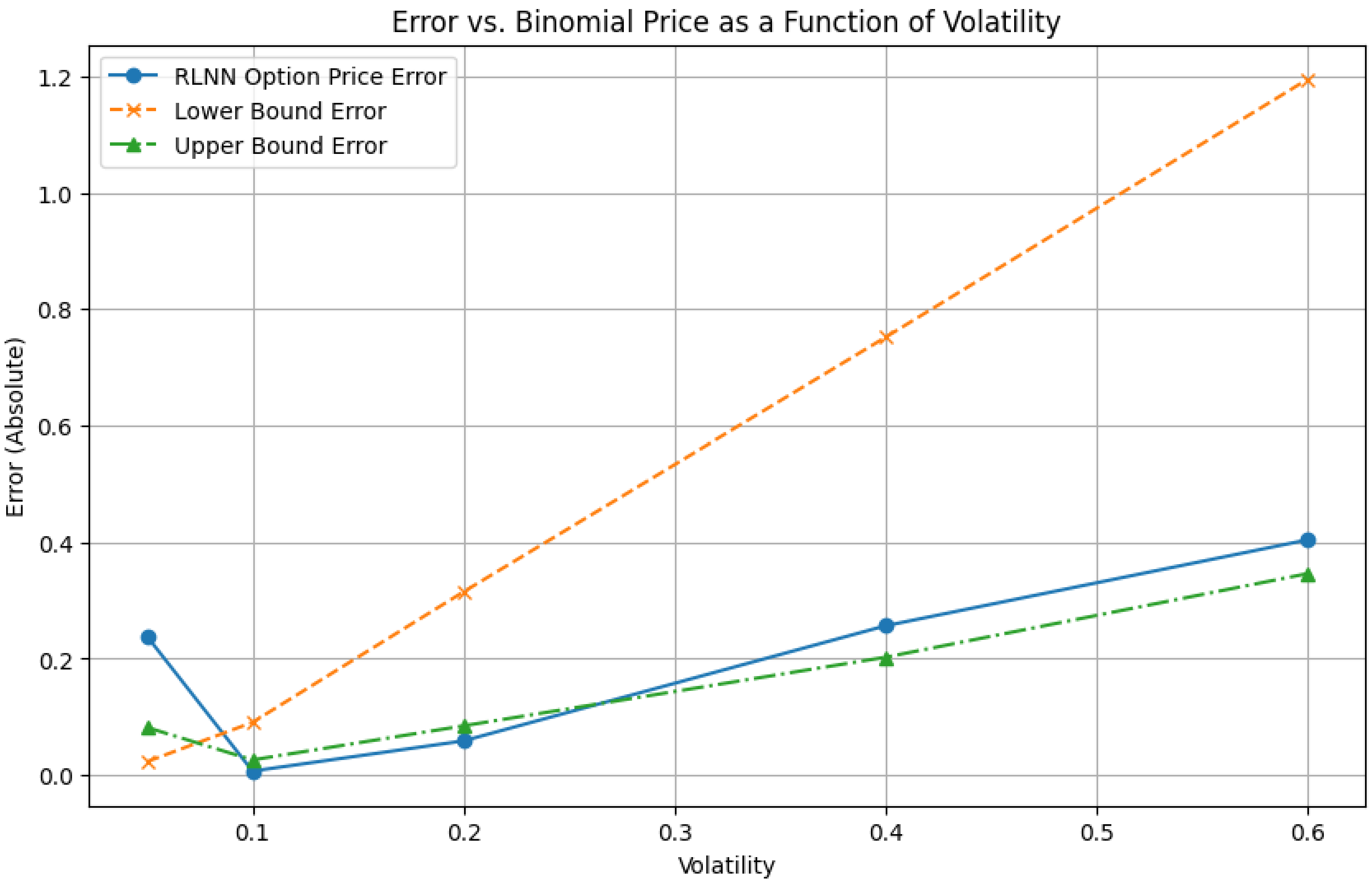
Parameters: S0=50, K=50, μ=0.06, σ=0.2, T=1, N=4, M=1500, N_L=20000

Results: Vega



Parameters: S0=50, K=50, μ=0.06, σ=0.2, T=1, N=4, M=1500, N_L=20000

Results: Vega



Parameters: S0=50, K=50, μ=0.06, σ=0.2, T=1, N=4, M=1500, N_L=20000

Delta Hedging

- **Step 1: Simulate Stock Paths**

$$S_{t+1}^m = S_t^m \exp\left((r - \frac{\sigma^2}{2})\Delta t + \sigma\sqrt{\Delta t} Z_t^m\right), \quad Z_t^m \sim \mathcal{N}(0,1)$$

- **Step 2: Compute Option Deltas**

$$\Delta_t = \frac{V_{t+1}^{\text{up}} - V_{t+1}^{\text{down}}}{S_{t+1}^{\text{up}} - S_{t+1}^{\text{down}}}$$

- **Step 3: Hedge Adjustments**

$$\text{Adjustment Cost} = (\Delta_t - \Delta_{t-1}) \cdot S_t$$

$$\text{Cost}_{\text{total}} = \sum_{t=1}^{N} (\Delta_t - \Delta_{t-1}) S_t e^{-r(T-t)}$$

- **Step 4: Profit and Loss (P&L)**

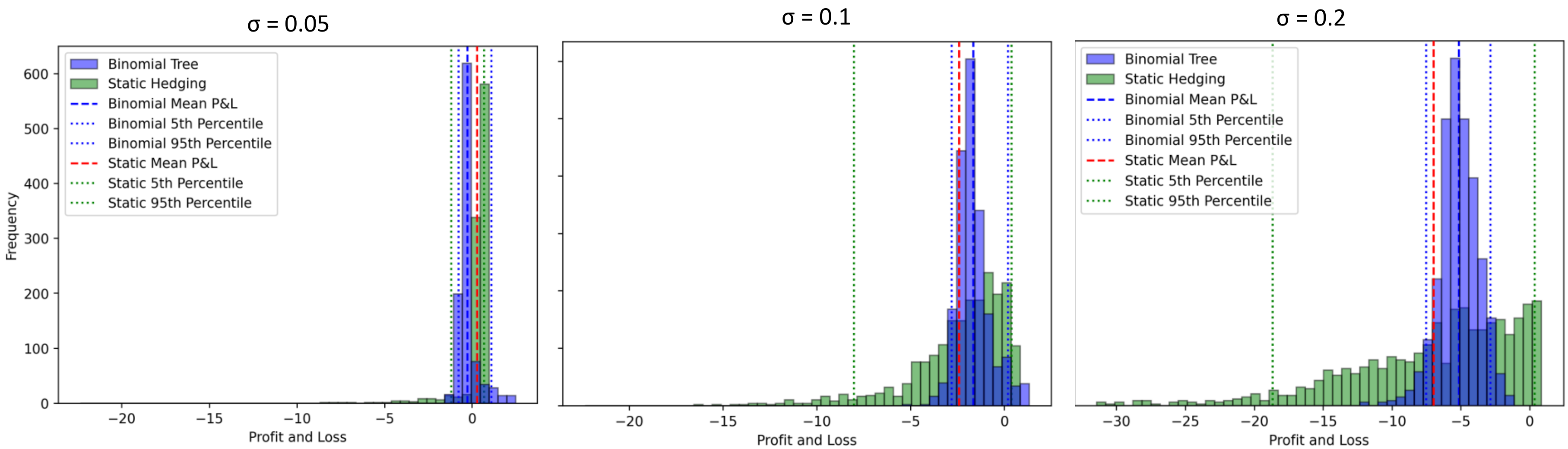
$$\text{P\&L} = \text{Payoff}_{\text{option}} - \text{Cost}_{\text{total}}$$

$$\mu = \frac{1}{M} \sum_{m=1}^{M} \text{P\&L}^m$$

Parameters: S0=50, K=50, μ=0.06, σ=0.2, T=1, N=4, M=10000

Results: Delta Hedging

Profit & Loss RLNN Statis Hedge vs. Binomial Tree Hedge



Parameters: S0=50, K=50, μ=0.06, σ=0.2, T=1, N=4, M=1000

Extensions

- **Multi asset derivatives**

Appendix: Further extensions if time allows

- **GPU implementation** (inc. computational efficiency assessment)
- **Backtesting** (assess feasibility of static hedge in practice)

Appendix: Bounds Computation

- **Lower Bound:**
  - Simulate new paths.
  - Use RLNN continuation values to determine stopping times.
  - Compute discounted payoffs:

$$\text{Lower Bound} = \frac{1}{N_L} \sum_{n=1}^{N_L} \frac{h(S_{\tau(n)})}{B_{\tau(n)}}$$

- **Upper Bound:**
  - Use dual formulation with martingales:

$$M_t = \sum_{i=0}^{m-1} \left(\frac{\tilde{G}_{\beta_{t_i+1}}(S_{t_i+1})}{B_{t_i+1}} - \frac{Q_{t_i}(S_{t_i})}{B_{t_i}} \right)$$

  - Construct the upper bound:

$$\text{Upper Bound} = \mathbb{E}\left[\max_t \left(\frac{h(S_t)}{B_t} - M_t \right)\right]$$

Advantages:

- **Efficient:** Avoids costly sub-simulations.
- **Accurate:** Tighter bounds for Bermudan-style derivatives.
- **Versatile:** Handles multi-asset, path-dependent claims effectively.

References

[1] Lokeshwar, V., Bharadwaj, V., & Jain, S. (2022). Explainable neural network for pricing and universal static hedging of contingent claims. *Applied Mathematics and Computation*, *417*, 126775.