Негосударственное образовательное учреждение
высшего образования

РОССИЙСКАЯ ЭКОНОМИЧЕСКАЯ ШКОЛА (институт)

Программа «Магистр экономики»

КУРСОВАЯ РАБОТА

TERM PAPER

Тема Открывая «черный ящик»: интерпретация стратегий глубокого хеджирования методом символьной регрессии

Title Opening the Black Box: Interpreting Deep Hedging Policies via Symbolic Regression

Студент/Student
Зерников Кирилл Геннадьевич
(ФИО студента, выполнившего работу)

Руководитель/Advisor
Горовой Вячеслав Сергеевич, PhD, профессор РЭШ
(ученая степень, звание, место работы, ФИО)

Оценка/Grade

_____

Подпись/Signature

_____

Москва, 2026

Abstract

The hedging of financial derivatives in markets characterized by frictions and stochastic volatility presents a significant challenge to traditional parametric models. This term paper investigates the application of Deep Reinforcement Learning (DRL) to the hedging of at-the-money call options, specifically utilizing the Twin Delayed Deep Deterministic Policy Gradient (TD3) algorithm. We train the agent in two distinct environments: a controlled Geometric Brownian Motion (GBM) setting and a realistic Heston Stochastic Volatility setting calibrated to S&P 500 data. Our results demonstrate that in the frictionless, constant-volatility regime, the DRL agent autonomously recovers the Black-Scholes hedging strategy without prior theoretical knowledge. In the stochastic volatility environment, the agent consistently outperforms the Black-Scholes benchmark, achieving lower portfolio variance and superior risk-adjusted returns. A key contribution of this work is the application of Symbolic Regression to interpret the "black-box" neural network policies. We find that the agent learns a closed-form hedging rule that structurally resembles the Black-Scholes delta but incorporates specific corrections for the leverage effect and fat-tailed return distributions. These findings suggest that Deep Hedging can effectively internalize complex market dynamics, and Symbolic Regression serves as a powerful tool for bridging the gap between machine learning performance and economic interpretability.

Contents

1 Introduction

The problem of hedging is one of the most important and common problems in finance. Rational market participants seek to preserve capital and minimize the variance of portfolio returns. And one of the key problems in hedging is the task of hedging an at-the-money call option with underlying asset.

Traditional hedging strategies, largely based on parametric models and the "Greeks," provide essential theoretical foundations but often falter when faced with real-world complexities such as dynamic volatility surfaces, discrete hedging intervals and transaction costs.

To address these limitations, recent literature has proposed "Deep Hedging" — a non-parametric approach that utilizes neural networks to learn optimal hedging policies. This term paper contributes to this growing field by focusing on the training and evaluation of neural-network-based models using purely simulated data and simulated data based on calibrating the Heston Stochastic volatility model.

The remainder of the term paper is organized as follows. In Section 2 we give literature review on the hedging of call options and recent advancements regarding application of state-of-the-art Reinforcement Learning techniques to this problem. Section 3 outlines the theoretical framework of "Deep Hedging", reviewing the application of Deep Reinforcement Learning to problems with Continuous Action Domain and derivation of reward structure. Then in section 4 we give implementation details regarding model that we have chosen. We proceed with section 5 and the results of Monte-Carlo experiments, in which we train our deep reinforcement learning (DRL) agent on synthetic data and compare it with the Black-Scholes delta hedging. In the same section we run Symbolic Regressions on the outputs of our agents and interpret policies that are implemented by them. Section 6 concludes and outlines the directions for future research.

2 Literature Review

The BSM model, pioneered by Black and Scholes in 1973 [1], is fundamentally rooted in the principle of delta hedging as a mechanism to eliminate the directional risk of a derivative relative to its underlying asset. However, the classical Black-Scholes-Merton framework relies on idealized assumptions, most notably the absence of jumps in returns and the existence of constant volatility. In practice and real-world applications, market participants are exposed to multifaceted risk factors that render single-instrument hedges inherently incomplete [2, 3, 4]. Furthermore, the theoretical requirement for continuous, cost-free rebalancing is unattainable in real-world markets. Even with high-frequency execution, transaction costs impose a physical limit on trading activity, inevitably leading to a residual balance of hedging errors and expenses. This discrepancy necessitates the development of a hedging procedure that can determine the most efficient path under realistic constraints.

From a modern computational perspective, the challenge of delta hedging can be reframed as a dynamic, sequential optimization problem [5]. The goal is to maximize a trader's expected cumulative utility by balancing the dual objectives of risk reduction and return maximization. This framing makes Reinforcement Learning (RL) an exceptionally suitable candidate for the task. The core objective of RL is to identify an optimal policy that dictates actions within an environment to maximize a discounted stream of cumulative rewards. In a financial context, these rewards are calibrated to penalize risk

while rewarding portfolio stability. A critical advantage of the RL approach is its focus on long-term outcomes; rather than seeking a not far-sighted, short-term fix, the algorithm optimizes for the total performance of the hedge over the entire life of the contract.

The operational mechanism of RL relies on an iterative process of trial and error, where an agent refines its strategy based on the feedback received from its interactions with the environment. Once a researcher defines the reward structure and the state parameters, the system autonomously learns to navigate the policy space. Perhaps the most significant breakthrough in this field is the synthesis of RL with deep neural networks. This combination allows the model to map optimal policies within high-dimensional state spaces—a requirement that is essential given the complexity of modern financial data [5, 6, 7]. By utilizing these deep learning architectures, the agent internalizes non-linear relationships and path-dependent market dynamics that traditional parametric models typically fail to capture. Such models have shown ability to consistently outperform traditional Black-Scholes delta hedging in different scenarios [5, 8, 9, 10, 11].

While deep learning and RL do demonstrate superior performance in stress tests and simulations, their "black-box" nature remains a significant obstacle for industry adoption. However, there is a lack of literature focusing on the interpretability issue of Deep Hedging. This term paper aims to fill this gap by applying the Symbolic Regression technique to Deep Hedging agents.

3 Theoretical Framework

3.1 Deep Reinforcement Learning with Continuous Action Domain

In the Reinforcement Learning (RL) framework, an agent learns to navigate an environment through a trial-and-error process aimed at maximizing a cumulative reward signal [12]. The system is composed of two interacting entities: the agent (the decision-maker) and the environment (everything external to the agent). The agent's objective is to determine an optimal policy that dictates actions based on environmental observations to maximize the long-term expected return [13].

RL problems generally categorize actions into discrete or continuous spaces. While discrete actions involve a finite set of choices, continuous action domains allow the agent to select any value within a defined range. In the context of derivative hedging, continuous actions are particularly suitable as they align with the assumption of fractional trading in a frictionless market. At each time step, an agent observes the state of the environment, $s_t \in \mathcal{S}$, and then selects an action $a_t \in \mathcal{A}$ with respect to its deterministic policy $\mu : \mathcal{S} \to \mathcal{A}$ with parameter vector $\phi \in \mathbb{R}^n$. We model this interaction as a Markov Decision Process (MDP). A stationary transition dynamics distribution defines the system, where the probability of the subsequent state depends only on the current state-action pair:

$$p(s_{t+1}|s_1, a_1, \ldots, s_t, a_t) = p(s_{t+1}|s_t, a_t)$$

The agent's performance over an episode, which in hedging terminates at the end of the episode or at the option's maturity, is measured by the total discounted reward from time t onwards:

$$\tilde{r}_t = \sum_{i=0}^{\infty} r_{t+i}\gamma^i$$

where $\gamma \in (0, 1]$ is the discount factor and $r_{t+i} = R(s_{t+i-1}, a_{t+i-1})$ represents the immediate reward. To formalize the optimization objective, we define the discounted state distribution $\rho^{\mu_\phi}(s'; t)$ as:

$$\rho^{\mu_\phi}(s'; t) := \int_{\mathcal{S}} \sum_{i=0}^{\infty} \gamma^{t+i} p_t(s) p(s \to s'; t+i, \mu_\phi) ds$$

where $p(s \to s'; t+i, \mu_\phi)$ is the probability of reaching state s' from state s after $t+i$ steps under the policy function μ_ϕ with parameters ϕ. The agent aims to maximize the expected return $J(\mu_\phi)$ under the policy μ_ϕ:

$$J(\mu_\phi) = \int_{\mathcal{S}} \rho^{\mu_\phi}(s; t) R(s, a) ds = \mathbb{E}_{s \sim \rho^{\mu_\phi}}[\tilde{r}_t | \mu_\phi],$$

where $a = \mu(s|\phi)$.

To achieve this in a data-driven manner, we utilize Temporal Difference (TD) methods, specifically Q-learning. The action-value function, $Q^{\mu_\phi}(s, a)$, represents the expected return starting from state s, taking action a, and following policy μ_ϕ thereafter:

$$\begin{aligned} Q^{\mu_\phi}(s, a) &= \mathbb{E}_{s \sim \rho^{\mu_\phi}}[\tilde{r}_t | s_t = s, a_t = a; \mu_\phi] \\ &= r(s_t, a_t) + \gamma \mathbb{E}_{s \sim \rho^{\mu_\phi}}[Q^{\mu_\phi}(s_{t+1}, \mu(s_{t+1}|\phi))] \end{aligned}$$

In environments with high-dimensional state spaces, the optimal function $Q^*(s, a) = \max_\phi Q^{\mu_\phi}(s, \mu(s|\phi))$ is approximated via deep neural networks $Q(s, a|\theta)$ with parameters θ.

The optimization of the policy is governed by the deterministic policy gradient theorem. For a deterministic policy μ_ϕ, the gradient of the performance objective is given by:

$$\begin{aligned} \nabla_\phi J(\mu_\phi) &= \int_{\mathcal{S}} \rho^{\mu_\phi}(s) \nabla_\phi \mu(s|\phi) \nabla_a Q_\theta(s, a)|_{a=\mu(s|\phi)} ds \\ &= \mathbb{E}_{s \sim \rho^{\mu_\phi}}[\nabla_\phi \mu(s|\phi) \nabla_a Q_\theta(s, a)|_{a=\mu(s|\phi)}] \end{aligned}$$

One of the most prominent algorithms for continuous control is the Deep Deterministic Policy Gradient (DDPG). It utilizes an actor-critic architecture where the critic minimizes a loss function $L(\theta_i)$ based on the temporal difference error:

$$L(\theta_i) = \mathbb{E}_{s_i, a_i, r_i, s_{i+1}}(Q(s_i, a_i|\theta_i) - y_i)^2$$

The target value y_i is computed using target networks to enhance stability:

$$y_i = r(s_i, a_i) + \gamma Q(s_{i+1}, a_i^*|\theta_i^*)$$

In practice, the critic is updated using a random minibatch of size N from the replay buffer $\mathcal{B}$:

$$L(i) = \frac{1}{N} \sum_j (Q(s_j, a_j|\theta_i) - y_j)^2$$

Stability is further maintained by slowly updating the target network parameters (θ^*, ϕ^*) via:

$$\theta_{i+1}^* \leftarrow \tau \theta_i^* + (1 - \tau)\theta_i, \quad \phi_{i+1}^* \leftarrow \tau \phi^* + (1 - \tau)\phi_i$$

Despite its popularity, DDPG often suffers from overestimation bias in the Q-values. To mitigate this, we implement the Twin Delayed Deep Deterministic Policy Gradient (TD3) algorithm. TD3 employs two independent critic networks and calculates the target value y_j using the minimum of the two:

$$y_j = r(s_j, a_j) + \gamma \min_{k=1,2} Q(s_{j+1}, a^*_{k,j+1} | \theta^*_{k,j})$$

where the target action $a^*_{k,j+1} = \mu(s_{j+1}|\theta^*_{k,j}) + \epsilon_{j+1}$ includes clipped noise to smooth the value surface. To ensure physical and economic consistency, the final action is restricted to the valid hedging range:

$$a_i = \text{clip}(\mu(s_i|\phi_i) + \epsilon_i, a_{\text{low}}, a_{\text{high}})$$

This robust framework allows the DRL agent to learn complex hedging strategies directly from market data without necessitating prior assumptions regarding volatility or underlying price processes.

3.2 Reward function

The definition of the reward function is the most critical component in reinforcement learning, as it implicitly defines the agent's objective and governs the trade-off between competing goals. In the context of derivatives hedging under market frictions, the agent faces a classic optimization problem: minimizing the variance of the portfolio's final wealth (hedging error) while simultaneously minimizing the transaction costs incurred to achieve that hedge.

Following the theoretical framework established by [14], we interpret the hedging problem through the lens of maximizing expected utility. Specifically, we consider an agent with constant absolute risk aversion (CARA) or, equivalently in the context of normal distributions, an agent maximizing a mean-variance objective. The global objective is to maximize the expected terminal wealth w_T penalised by its variance:

$$\max_{\pi} \left(\mathbb{E}[w_T] - \frac{\kappa}{2} \mathbb{V}[w_T] \right),$$

where κ represents the coefficient of risk aversion and maximization is done over all possible strategies π. As derived in [14], if the wealth increments are assumed to be independent across time steps—a reasonable approximation in random walk markets—the global variance decomposes into the sum of local variances. Consequently, the cumulative reward maximization in RL becomes equivalent to solving this global optimization problem if the immediate reward r_t at each step is defined as the risk-adjusted wealth increment:

$$r_t = \delta w_t - \frac{\kappa}{2} (\delta w_t)^2.$$

This theoretical insight confirms that training an RL agent with step-wise risk-adjusted rewards is mathematically consistent with training an expected-utility maximizer.

While the quadratic variance penalty is theoretically elegant, empirical studies such as [8] and [10] suggest that using Standard Deviation (SD) rather than variance often leads to more stable convergence during the training of deep neural networks. Therefore, while we build upon the "automatic hedging" logic of [14], we formulate our specific objective to maximize the expected wealth minus a penalty proportional to the standard deviation:

$$J(\pi) = \mathbb{E}[w_T] - \xi \text{SD}(w_T),$$

where $\xi > 0$ is the risk-aversion parameter controlling the trade-off. So we effectively move to Mean-Absolute Deviation optimization, which is a proxy for Mean-Variance. To approximate this in a step-wise fashion suitable for the TD3 algorithm, we define the immediate reward r_t as:

$$r_t = \mathrm{PnL}_t - \xi|\mathrm{PnL}_t|, \tag{3.1}$$

where PnL_t represents the change in the agent's total portfolio wealth between time $t-1$ and t. This formulation encourages the agent to generate positive returns while heavily penalizing large magnitude fluctuations (volatility) in portfolio value.

The period profit and loss, PnL_t, is composed of the changes in the value of the option position, the underlying asset holdings, and the costs incurred from rebalancing. It is defined as:

$$\mathrm{PnL}_t = \Delta V_t^O + \Delta V_t^S - \mathrm{Costs}_t,$$

or more explicitly:

$$\mathrm{PnL}_t = H_t^O(C_t - C_{t-1}) + H_t^S(S_t - S_{t-1}) - c|S_t(H_t^S - H_{t-1}^S)|,$$

where H_t^O is the option position (e.g., -1 for a short call); C_t and S_t are the prices of the option and the underlying asset at time t, respectively; H_t^S is the number of shares of the underlying asset held at time t; c represents the proportional transaction cost parameter.

In this framework, the transaction cost term creates the central friction. If $c = 0$, the agent could theoretically approach the continuous Black-Scholes hedge (perfect replication). However, in the presence of significant transaction costs, the DRL agent is expected to outperform the Black-Scholes benchmark, as it has the ability to "wait" or under-hedge to avoid eroding wealth through excessive trading. To isolate model risk from friction management, in our study c is set conservatively to 1 basis point and we measure relative performance of our agent by comparing it with the Black-Sholes delta hedging benchmark.

4 Implementation details

Our implementation of the TD3 algorithm very closely follows the one in [10]. Like in their paper, we define the state observation vector to include four primary features: moneyness, time to maturity, the agent's current inventory (holdings), and the Black-Scholes implied volatility. It is important to note that we explicitly exclude analytical Greeks, such as Delta or Gamma, from the state space. By withholding these theoretical derivatives, we force the neural network to autonomously "learn the Greeks" and the underlying hedging dynamics solely through trial-and-error interactions with the data. However, including implied volatility is essential, as it provides the agent with the same information set used in standard pricing models, allowing the neural network to theoretically recover the Black-Scholes benchmark as a special case.

Featurization and neural network architecture are exactly the same as in [10], i.e. z-score normalization is applied to all inputs, TD3 algorithm is implemented using feed-forward neural networks for both the Actor (policy) and the Critic (value estimator). The Actor network comprises two hidden layers, each containing 256 neurons. The output layer utilizes a Hyperbolic Tangent (tanh) activation function, which is scaled to constrain the agent's actions within the allowable hedging limits (e.g., $[0, 1]$). The Critic network is slightly deeper, consisting of three hidden layers of approximately 250 neurons

each. For hidden layers Leaky ReLU activation function with a leakiness parameter of $\alpha = 0.05$ is employed. The network weights are optimized using the Adam optimizer with a conservative learning rate of 10^{-4} and a relatively large batch size of 1000, which contributes to the stability of the learning process.

Again similarly to [10], we inject Gaussian noise into the agent's actions to manage the exploration-exploitation trade-off. The magnitude of this noise is high at the beginning of training to encourage broad exploration of the action space and decays as the policy converges. The TD3-specific soft update parameter, which controls how establishing the target networks track the learned networks, is set to $\tau = 0.001$, ensuring smooth and stable policy updates.

The main difference from [10] is that in our case hedging environment is modeled in discrete time with daily rebalancing frequency and a single training episode is defined as a 21-day period. We also use different data for calibrating Heston model, see below.

5 Data and Results

5.1 Data

To calibrate the Heston model, we use S&P 500 index daily data between January 2$^{\text{nd}}$ 2019 and December 29$^{\text{th}}$ 2021, provided by optionsDX.[1] For each option quote the following information is available: the value of the underlying index, strike, maturity date, call-put flag, bid price, bid quantity, ask price, ask quantity, as well as greeks and implied volatility. The P&L in 3.2 is calculated from the mid-price of the option. The risk free rate used is the 1 year point of the U.S. Treasury Yield Curve, obtained from the U.S. Department of the Treasury.[2]

Options that did not have any of the above mentioned parameters were removed, as well as options with zero prices, zero strikes and negative bid-ask spreads. For calibration only options with more than 7 but less than 90 days to expiry were chosen and with moneyness $0.85 \leq \frac{S}{K} \leq 1.15$. This is a standard practice in literature, as change in the delta of the option is highest when gamma is at its largest, i.e. when the option is at-the-money.

For the assessment of the benchmark and agent we have picked 5 metrics:

1) Mean episode P&L as the average of the total P&L over each 21-day episode

2) Standard episode P&L as the standard deviation of the episode total P&Ls over the 10,000 runs

3) Mean episode transaction costs over each 21-day episode

4) Average rewards accumulated as defined in eq. 3.1

5) CVaR 5% is the average of the worst 5% of episodes in terms of P&L

[1] https://www.optionsdx.com/product/spx-option-chain/

[2] https://home.treasury.gov/policy-issues/financing-the-government/interest-rate-statistics?data=yield

Table 1: Summary statistics on the estimated parameters obtained from daily calibrations of the Heston (1993) model. We use the following notation for Heston model:

$$dS_t = \mu S_t dt + \sqrt{v_t} S_t dW_{1,t},$$
$$dv_t = \kappa(v_t - \theta)dt + \eta\sqrt{v_t} dW_{2,t},$$

where $\kappa > 0, \theta > 0, \eta > 0$, and $dW_{1,t} dW_{2,t} = \rho dt, \rho \in [-1, 1]$.

	θ	κ	η	ρ	v_0
Min	0.02	0.14	0.01	-0.99	0.01
Max	0.50	20.00	2.92	-0.00	0.50
Median	0.11	3.08	1.05	-0.88	0.02
Average	0.23	4.91	1.04	-0.88	0.04
Standard Deviation	0.20	4.91	0.39	0.10	0.07

5.2 Constant volatility

In the first part of our Monte Carlo experiment, we generate 20,000 hedging episodes (one episode lasts for 21 days) from a classic model of Geometric Brownian Motion with volatility parameter $\sigma = 0.25$, which we use to train our DRL agent. Then we test the agent on another set of synthetic data, generated with the same parameters, that consists of 10,000 episodes. For each episode agent hedges an option with moneyness from uniform distribution on [0.85; 1.15] and with time to maturity from uniform distribution on [40 days, 90 days]. Note that under low transaction costs and constant volatility Black-Scholes delta hedging theoretically should be very close to optimal. Thus in this part we would like to ensure that our agent can demonstrate satisfactory performance in a setting that clearly favours standard Black-Scholes delta hedging procedure. Table 2 reports results of testing our agent in this setting for $\xi \in \{1, 2, 3\}$.

Table 2: Deep reinforcement learning (DRL) agent's performance and hedging cost against the classic Black-Scholes delta hedging benchmark under constant volatility (GBM model). The results are reported for $\xi = 1, 2, 3$. Greater ξ gives more weight to risk minimization.

	Mean episode P&L	Std episode P&L	CVaR 5%	Mean episode transaction costs	Rewards
Panel A: $\xi = 1$					
Black-Scholes Delta Hedging	-0.0051 %	0.2287 %	-0.4249 %	-0.0116 %	**-0.39**
DRL Agent	-0.0048 %	0.2302 %	-0.4229 %	-0.0117 %	-0.51
Panel B: $\xi = 2$					
Black-Scholes Delta Hedging	-0.0049 %	0.2270 %	-0.4301 %	-0.0117 %	**-7.02**
DRL Agent	-0.0031 %	0.2354 %	-0.4455 %	-0.0117 %	-8.09
Panel C: $\xi = 3$					
Black-Scholes Delta Hedging	-0.0054 %	0.2246 %	-0.4246 %	-0.0117 %	**-13.67**
DRL Agent	-0.0074 %	0.2336 %	-0.4436 %	-0.0117 %	-14.96

Indeed we can see that performance of the DRL agent is almost as good as the one of practitioner's delta hedging, which is further supported by PnL distribution presented in Figure 1 in section A.1 of Supplementary Figures. This strongly suggests that our agent was able to recover the classical delta hedging on its own. However, policy implemented by our agent remains opaque due to the black-box nature of deep neural networks.

To interpret this "black box" policy learned by the neural network, we applied Symbolic Regression using the `PySR` library. We generated a probe dataset of $N = 5000$ state-action pairs from the trained agent, covering a moneyness range [0.85, 1.15] and time to maturity [40, 90] days. To isolate the agent's pricing logic from its transaction cost management, we queried the agent by initializing it with the theoretically optimal (Black-Scholes) position. This effectively removes the friction penalty, allowing us to observe the agent's "target" hedge ratio δ_{GBM}.

The inputs provided to the symbolic regressor were Moneyness $M = S/K$ and Time to Maturity τ (in years). The target variable was the agent's output action (hedge ratio). We restricted the search space to basic arithmetic operators plus $\ln(\cdot)$, $\sqrt{\cdot}$, and the error function $\text{erf}(\cdot)$, which is defined as:

$$\text{erf}(z) = \frac{2}{\sqrt{\pi}} \int_0^z e^{-t^2} dt = 2\Phi(\sqrt{2}z) - 1,$$

where $\Phi(\cdot)$ is the standard normal cumulative distribution function.

The symbolic regression algorithm converged to the following analytical expression for the agent's policy δ_{GBM}:

$$\delta^{SR}_{\text{GBM}}(M, \tau) = 0.494 + 0.507 \cdot \text{erf}\left(\frac{2.814 \cdot \ln(M)}{\sqrt{\tau}}\right) + 0.100 \cdot \tau \tag{5.1}$$

Note that this regression provides a very good and robust fit to the δ_{GBM}, as shown in Figure 1 in section A.1 and in Figure 3 in section A.3 of Supplementary Figures. This result is remarkably consistent with the theoretical Black-Scholes Delta definition. Recall that the Delta of a European Call option is given by $N(d_1)$, which can be expanded using the error function:

$$\Delta_{BS} = \frac{1}{2} + \frac{1}{2} \cdot \text{erf}\left(\frac{d_1}{\sqrt{2}}\right)$$

Approximating $d_1 \approx \frac{\ln(M)}{\sigma\sqrt{\tau}}$ (assuming small drift, which it is in our simulation), the theoretical equation becomes:

$$\Delta_{BS} \approx 0.5 + 0.5 \cdot \text{erf}\left(\left[\frac{1}{\sigma\sqrt{2}}\right] \frac{\ln(M)}{\sqrt{\tau}}\right)$$

Comparing Equation (5.1) with the theoretical model reveals a near-perfect structural match. The agent identified the sigmoid activation dynamics inherent in option pricing via the error function. Furthermore, a quantitative analysis of the coefficients demonstrates that the agent correctly inferred the latent volatility parameter σ used in the Geometric Brownian Motion simulation ($\sigma = 0.25$).

The theoretical scaling factor inside the error function is:

$$k_{theory} = \frac{1}{\sigma\sqrt{2}} = \frac{1}{0.25 \cdot 1.414} \approx 2.828$$

The factor recovered from the neural network is $k_{agent} = 2.814$. The relative error is merely 0.5%. This result provides strong empirical evidence that the Deep Hedging agent successfully "learned" the Black-Scholes model from scratch, purely through trial-and-error interaction with the stochastic environment, without any prior knowledge of the differential equations governing the system.

5.3 Stochastic Volatility

In the second part of our Monte Carlo experiment, we generate data from a Heston model in the following way. For each training episode we pick a random day between January 2nd 2019 and December 29th 2021 for which we have calibrated a Heston model. Then we simulate 21 days with these parameters via the standard Heston model and on this synthetic data train our agent. Training consists of 20000 episodes and testing is done on 10000 out-of-sample episodes. Similarly to the case of constant volatility, for each episode agent hedges an option with moneyness from uniform distribution on [0.85; 1.15] and with time to maturity from uniform distribution on [40 days, 90 days]. Note that in this scenario, when correlation between returns and volatility of underlying is non-zero, classic Black-Scholes delta hedging is not optimal even theoretically and even under zero transaction costs. So in this case we expect our agent to actually beat Black-Scholes benchmark, which is confirmed by the metrics in Table 3. We can also clearly see that this is achieved by lower variance and reduction in CVaR, for PnL distribution see Figure 2 in section A.2 of Supplementary Figures.

Table 3: Deep reinforcement learning (DRL) agent's performance and hedging cost against the classic Black-Scholes delta hedging benchmark under stochastic volatility (Heston model). The results are reported for $\xi = 1, 2, 3$. Greater ξ gives more weight to risk minimization.

	Mean episode P&L	Std episode P&L	CVaR 5%	Mean episode transaction costs	Rewards
Panel A: $\xi = 1$					
Black-Scholes Delta Hedging	-0.1835 %	0.4922 %	-1.1459 %	-0.0085 %	-9.64
DRL Agent	-0.1771 %	0.4042 %	-1.0474 %	-0.0075 %	**-6.77**
Panel B: $\xi = 2$					
Black-Scholes Delta Hedging	-0.3043 %	0.6103 %	-1.5735 %	-0.0116 %	-38.64
DRL Agent	-0.3034 %	0.5020 %	-1.4267 %	-0.0104 %	**-29.89**
Panel C: $\xi = 3$					
Black-Scholes Delta Hedging	-0.2817 %	0.6374 %	-1.5638 %	-0.0113 %	-59.69
DRL Agent	-0.2819 %	0.5280 %	-1.4524 %	-0.0105 %	**-47.28**

Following the successful replication of the GBM case, we extended the analysis to this Stochastic Volatility model case. Unlike the GBM case where volatility is constant, the Heston model introduces time-varying, stochastic volatility, significantly increasing the complexity of the optimal hedging surface. Also note that our agent was not trained on a single parametrization of the Heston model; the setting in which it was trained is effectively a combination of multiple Heston models with different parameters (see Table 1).

Similarly to Constant Volatility case, we queried the trained actor network on a synthetic dataset of $N = 5000$ state-action pairs, covering a moneyness range $M \in [0.85, 1.15]$, time to maturity $\tau \in [40, 90]$ days and volatility $v \in [0.10, 0.40]$. We employed symbolic regression to search for a simplified functional mapping from the state variables $\mathbf{x} = [M, \tau, v]$ to the hedge ratio $\delta_{\text{Stochastic}}$.

Remarkably, without any prior knowledge of option pricing theory, the symbolic regression algorithm recovered a functional form striking similar to the Black-Scholes-Merton equation. The best-fit expression discovered is:

$$\delta^{SR}_{Stochastic}(M, \tau, v) = 0.489 + 0.507 \cdot \text{erf}\left(\frac{0.640 \cdot (M - 1.014)}{v\sqrt{\tau}}\right) \tag{5.2}$$

For check of goodness-of-fit see Figure 4 in section A.4 of Supplementary Figures.

This result is significant for several reasons. First, the agent successfully identified the non-linear dependency on time and volatility. Specifically, it "discovered" the fundamental diffusion scaling law, placing the volatility and the square root of time in the denominator of the argument:

$$\text{Argument}_{SR} \propto \frac{M - 1}{v\sqrt{\tau}}$$

This mirrors the structure of the standardized moneyness d_1 in the Black-Scholes model, where $d_1 \approx \frac{\ln(M)}{v\sqrt{\tau}}$.

While the functional structure aligns with standard theory, the coefficients reveal specific behavioral adaptations learned by the agent to cope with stochastic volatility.

As shown in Table 4, the agent consistently outputs a hedge ratio lower than the Black-Scholes benchmark, which is further supported by delta correction surfaces in Figure 3. Our Symbolic Policy Distillation reveals that the Deep RL agent trained under the Heston process does not merely approximate the Black-Scholes model but fundamentally corrects it. While the recovered symbolic functional form (Equation 5.2) retains the sigmoid structure of the Normal CDF, the deviations in its parameters reveal that the agent has learned to account for the Leverage Effect inherent in stochastic volatility dynamics.

Table 4: Comparison of Theoretical Delta vs. Deep Agent vs. Symbolic Approximation (Selected Samples)

Moneyness (M)	Days (τ)	Vol (v)	Δ_{BS}	$\delta_{\text{Stochastic}}$	Diff
1.109	82.3	0.272	0.8023	0.7265	-0.075
1.012	73.2	0.111	0.5851	0.4391	-0.146
0.946	41.6	0.318	0.3165	0.2706	-0.046
1.087	44.8	0.134	0.9621	0.9035	-0.059
0.928	59.2	0.212	0.1957	0.1652	-0.030

In the Heston model, the dynamics of the asset price S_t and variance v_t are correlated with coefficient ρ. In equity markets, typically $\rho < 0$. This implies that upward shocks to the asset price are associated with downward shocks to volatility.

The Black-Scholes delta, $\Delta_{BS} = \frac{\partial C}{\partial S}$, assumes volatility is independent of price. However, the true sensitivity of the option price in this environment is:

$$dC \approx \frac{\partial C}{\partial S} dS + \frac{\partial C}{\partial v} dv$$

Substituting the expected relationship $dv \approx \rho \frac{\sigma_v \sqrt{v}}{S\sqrt{v}} dS$, the effective hedging ratio becomes:

$$\Delta_{Total} \approx \Delta_{BS} + \rho \left(\frac{\sigma_v}{S} \right) \mathcal{V}_{ega}$$

Since Call options have positive Vega ($\mathcal{V}_{ega} > 0$) and $\rho < 0$, the correction term is negative. The Deep RL agent, driven by a reward function that penalizes variance ($\kappa \text{Var}[PnL]$), naturally discovers this relationship. It learns that simply holding Δ_{BS} shares over-hedges the position because it fails to account for the volatility crash associated with rising prices.

The recovered symbolic expression provides further evidence of this structural learning:

$$\text{Argument}_{SR} = \frac{0.640 \cdot (M - 1.014)}{v\sqrt{\tau}}$$

The scaling coefficient 0.640 is notably smaller than the theoretical Black-Scholes coefficient of $1/\sqrt{2} \approx 0.707$. This flattening of the sigmoid curve (lower Gamma) is a rational response to the "Fat Tails" (excess kurtosis) of the Heston process. In a regime where extreme moves are more probable than in a Gaussian world, a risk-averse agent minimizes the rebalancing frequency and magnitude to avoid amplifying losses during high-volatility oscillations.

In summary, the agent has autonomously derived a sophisticated hedging strategy that integrates Delta and Vega risks into a single execution instrument, effectively correcting the Black-Scholes model for the reality of stochastic volatility.

6 Conclusion

In this term paper, we explored the capabilities of Deep Reinforcement Learning to solve the dynamic hedging problem for European call options. By implementing a TD3-based agent, we sought to determine whether a model-free approach could not only replicate established theoretical benchmarks but also improve upon them in complex market environments.

Our analysis yielded two primary conclusions. Firstly, in the case of constant volatility, the agent demonstrated the ability to learn the optimal Black-Scholes hedging strategy purely from trial-and-error interactions with synthetic data. The application of Symbolic Regression confirmed this by recovering a functional form nearly identical to the analytical Black-Scholes Delta, utilizing the error function to approximate the cumulative normal distribution. This serves as a robust validation of the DRL framework's capacity to identify fundamental financial laws.

Secondly, under the Heston Stochastic Volatility model, the Deep Hedging agent outperformed the traditional delta-hedging benchmark. The agent achieved a reduction in the standard deviation of the terminal P&L and improved the conditional value at risk (CVaR). Through Symbolic Policy Distillation, we revealed that the agent adopted a strategy of systematic under-hedging relative to the Black-Scholes delta. This behaviour is economically rational, as it accounts for the negative correlation between asset returns and volatility (the leverage effect) and the presence of transaction costs, factors that standard parametric models often neglect or oversimplify.

Even though there are several promising avenues for future research like extending the presented framework to path-dependent exotics or incorporating Long Short-Term Memory (LSTM) networks into the actor-critic architecture, there is one particular area

that we hope to explore in the future. The main advantage of following implementation presented in [10] is the fact that authors were able to train their agent on purely empirical data, without any assumptions on model dynamics. We hope to extend our Symbolic Regression analysis to a model trained exclusively on empirical data and check how it differs from basic Black-Scholes benchmark and from the model trained in Heston world.

A Supplementary Figures

A.1 PnL Distribution Analysis of the Constant Volatility case

Figure 1 illustrates the distribution of total Profit and Loss (PnL) per episode for the GBM model simulation. The distribution compares the Deep Hedging agent against the Black-Scholes benchmark over 10000 test episodes.

Shapes of Pnl distributions of agent (blue) and BS benchmark (dashed orange) are nearly identical, which illustrates that agent managed to successfully learn the optimal hedge in Black-Scholes world. Distributions for other ξ look similar.

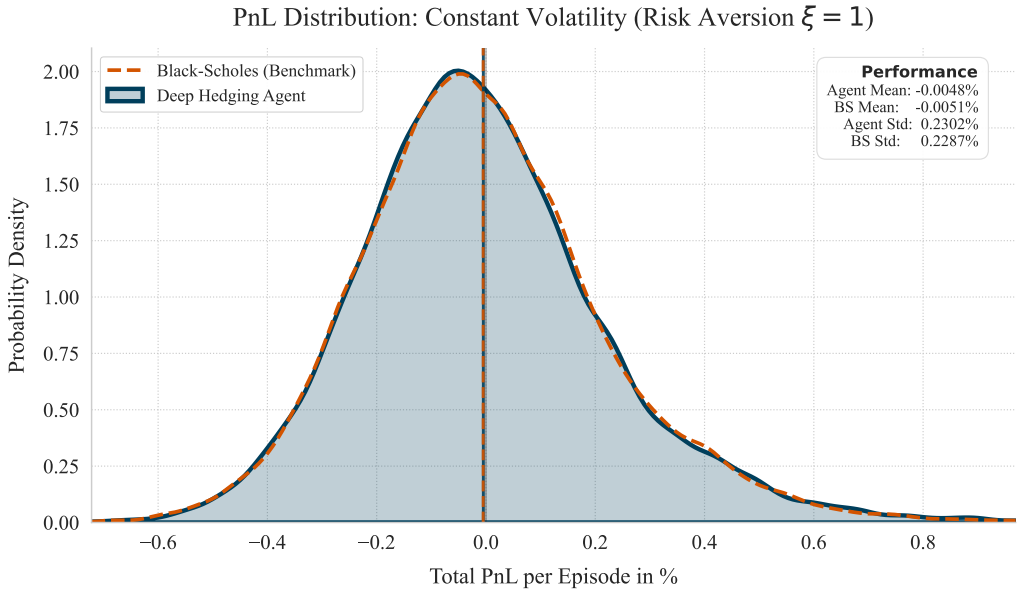


Figure 1: PnL Distribution Comparison (Constant Volatility). The density plot shows the aggregated PnL per episode for the Deep Hedging Agent vs. the Black-Scholes Benchmark. The text box summarizes the mean and standard deviation for both strategies.

A.2 PnL Distribution Analysis of the Stochastic Volatility Case

Figure 2 illustrates the distribution of total Profit and Loss (PnL) per episode for the Heston model simulation. The distribution compares the Deep Hedging agent against the Black-Scholes benchmark over 10000 test episodes.

The agent's PnL distribution (blue) demonstrates a significantly lower variance compared to the benchmark (dashed orange), indicating a more robust hedging strategy under stochastic volatility conditions. The peaks of both distributions are centered near zero, confirming that both strategies effectively hedge the risk on average, but the agent achieves tighter tails (lower kurtosis). Distributions for other ξ look similar.

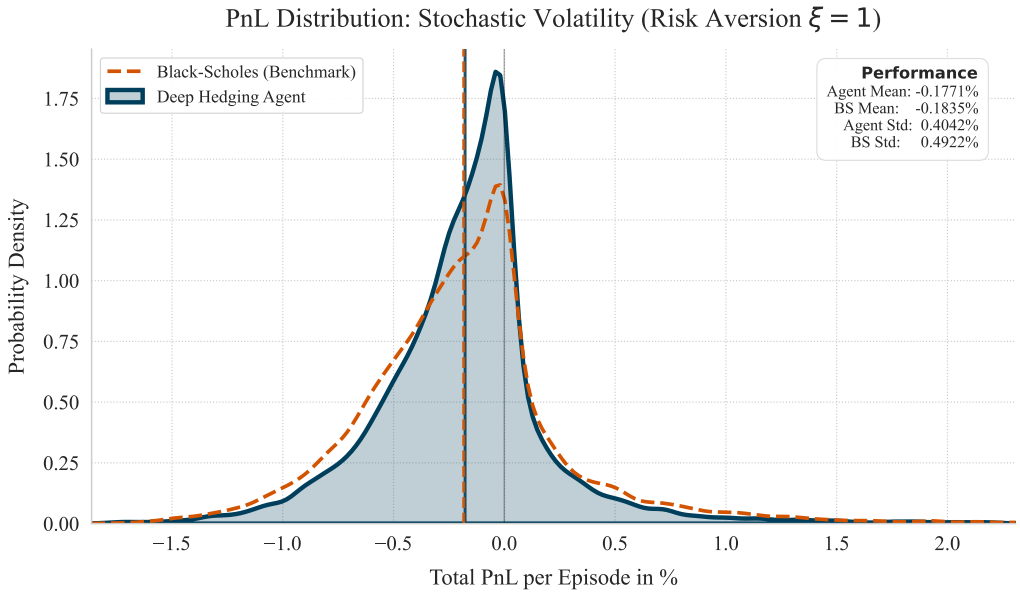


Figure 2: PnL Distribution Comparison (Stochastic Volatility). The density plot shows the aggregated PnL per episode for the Deep Hedging Agent vs. the Black-Scholes Benchmark. The text box summarizes the mean and standard deviation for both strategies.

A.3 Delta Surfaces

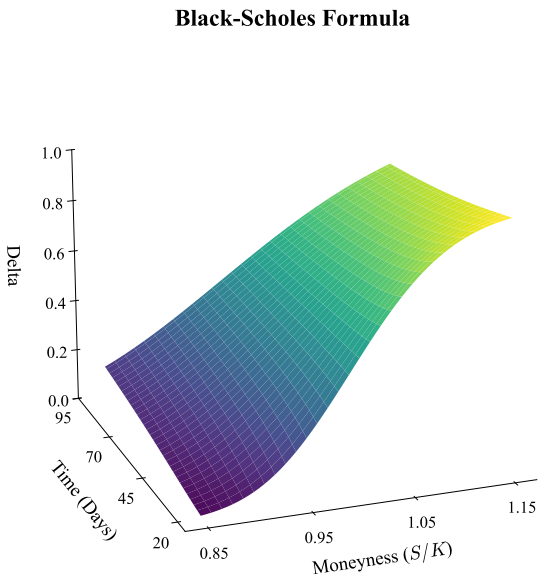
Figure 3 represents delta surfaces, i.e. dependency of delta on moneyness and time to maturity, of Black-Scholes formula, TD3 agent and formula distilled by symbolic regression. These three surfaces are effectively identical, which further supports the idea that out agent was able to successfully learn Black-Scholes formula for hedging and that we were able to recover from symbolic regression on the actions of agent.

A.4 Goodness-of-fit of Symbolic Regression in Stochastic Case

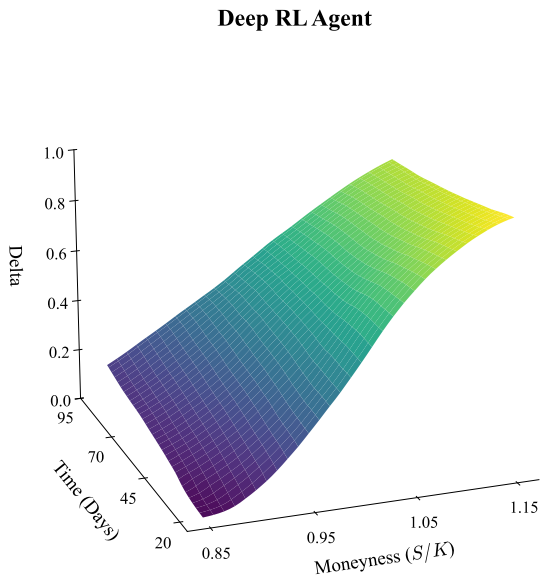
Figure 4 compares the hedge ratio predicted by the Symbolic Regression formula (δ_{SR}) against the actual output of the Deep RL agent (δ_{Agent}) across $N = 1000$ random samples per volatility regime. The tight alignment along the 45° dashed diagonal line ($y = x$) and the near-unity coefficients of determination ($R^2 > 0.99$) confirm that the compact symbolic equation provides a statistically robust approximation of the neural network's decision manifold.

A.5 Under-hedging in Stochastic Volatility case

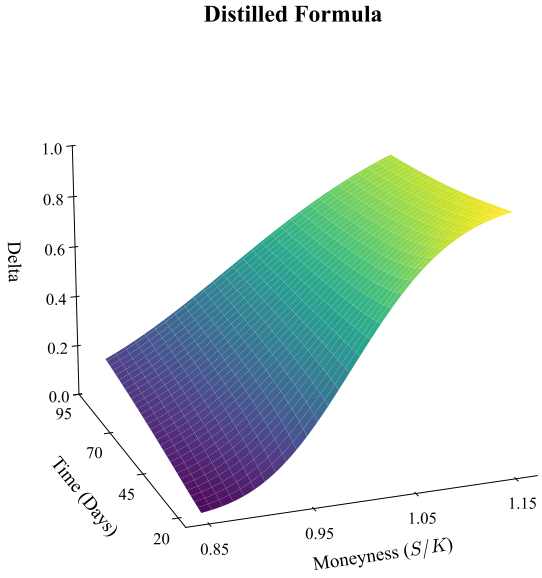
Figure 5 shows delta correction surfaces ($\Delta_{Agent} - \Delta_{BS}$) for three different volatility regimes. As we can see agent consistently under-hedges relative to Black-Scholes, a behaviour consistent with the minimum variance hedge in the presence of negative asset-volatility correlation (leverage effect).



(a) Ground Truth (Black-Scholes)

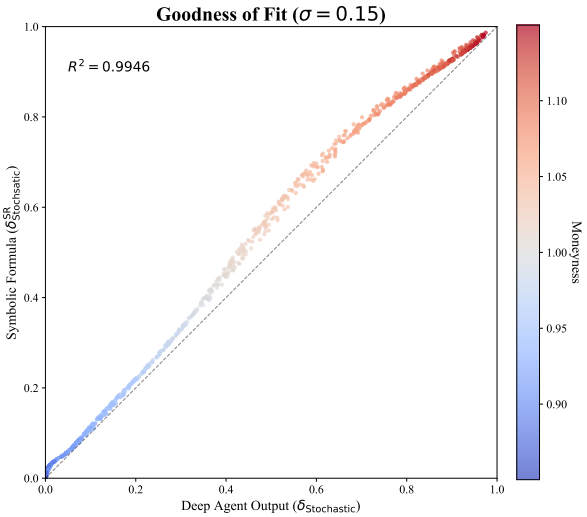


(b) Deep RL Agent (TD3)

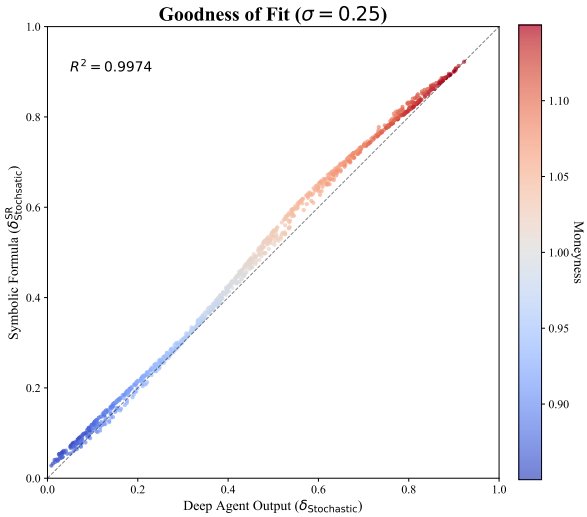


(c) Distilled Policy (Symbolic Regression)

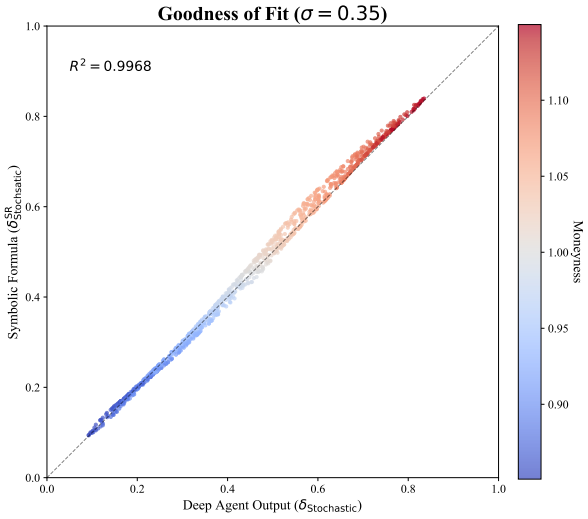
Figure 3: Delta Surfaces Comparison. The Deep RL agent (b) and the recovered symbolic formula (c) accurately reproduce the theoretical Black-Scholes surface (a) across varying moneyness and time-to-maturity.



(a) Low Volatility ($\sigma = 0.15$)

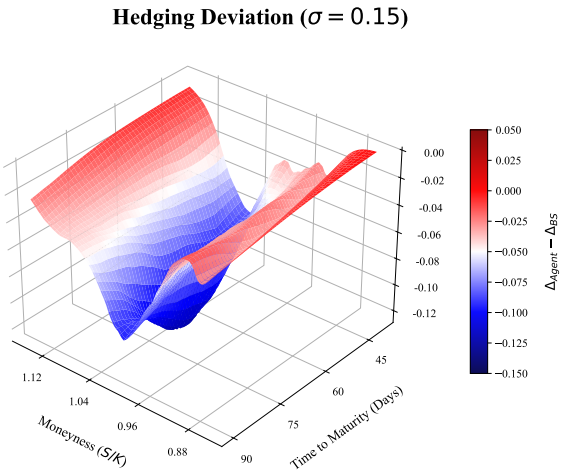


(b) Medium Volatility ($\sigma = 0.25$)

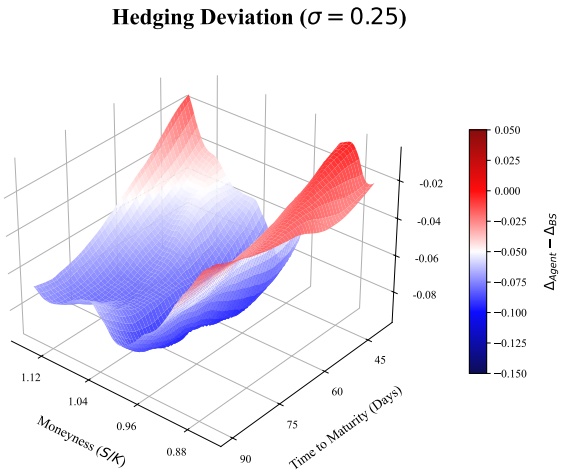


(c) High Volatility ($\sigma = 0.35$)

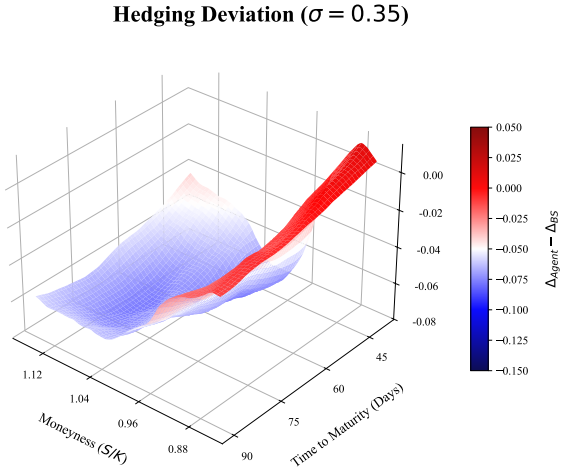
Figure 4: Goodness of Fit (Parity Analysis) Comparison of the hedge ratio predicted by the Symbolic Regression formula (δ_{SR}) against the actual output of the Deep RL agent (δ_{Agent}) across $N = 1000$ random samples per volatility regime. The points are colored by Moneyness (S/K).



(a) Ground Truth (Black-Scholes)



(b) Deep RL Agent (TD3)



(c) Distilled Policy (Symbolic Regression)

Figure 5: Delta Correction Surface ($\Delta_{Agent} - \Delta_{BS}$). The plots illustrate the deviation of the learned hedging strategy from the Black-Scholes benchmark across different volatility regimes $\sigma \in \{0.15, 0.25, 0.35\}$. The negative values (blue) indicate under-hedging of agent relative to Black-Scholes formula, while positive values (red) indicate over-hedging.

References

[1] F. Black and M. Scholes, "The pricing of options and corporate liabilities," *Journal of political economy*, vol. 81, no. 3, pp. 637–654, 1973.

[2] G. Bakshi, C. Cao, and Z. Chen, "Empirical performance of alternative option pricing models," *The Journal of finance*, vol. 52, no. 5, pp. 2003–2049, 1997.

[3] V. Naik, "Option valuation and hedging strategies with jumps in the volatility of asset returns," *The Journal of Finance*, vol. 48, no. 5, pp. 1969–1984, 1993.

[4] P. Tankov and E. Voltchkova, "Jump-diffusion models: a practitioner's guide," *Banque et Marchés*, vol. 99, no. 1, p. 24, 2009.

[5] H. Buehler, L. Gonon, J. Teichmann, and B. Wood, "Deep hedging," *Quantitative Finance*, vol. 19, no. 8, pp. 1271–1291, 2019.

[6] J. B. Henderson, K. L. McNeill, M. González-Howard, K. Close, and M. Evans, "Key challenges and future directions for educational research on scientific argumentation," *Journal of Research in Science Teaching*, vol. 55, no. 1, pp. 5–18, 2018.

[7] V. François-Lavet, P. Henderson, R. Islam, M. G. Bellemare, J. Pineau, *et al.*, "An introduction to deep reinforcement learning," *Foundations and Trends® in Machine Learning*, vol. 11, no. 3-4, pp. 219–354, 2018.

[8] J. Cao, J. Chen, J. Hull, and Z. Poulos, "Deep hedging of derivatives using reinforcement learning," *arXiv preprint arXiv:2103.16409*, 2021.

[9] A. Giurca and S. Borovkova, "Delta hedging of derivatives using deep reinforcement learning," *Available at SSRN 3847272*, 2021.

[10] O. Mikkilä and J. Kanniainen, "Empirical deep hedging," *Quantitative Finance*, vol. 23, no. 1, pp. 111–122, 2023.

[11] E. Huang and Y. Lawryshyn, "Deep hedging under market frictions: A comparison of drl models for options hedging with impact and transaction costs," *Journal of Risk and Financial Management*, vol. 18, no. 9, p. 497, 2025.

[12] A. G. Barto, "Reinforcement learning: An introduction. by richard's sutton," *SIAM Rev*, vol. 6, no. 2, p. 423, 2021.

[13] D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra, and M. Riedmiller, "Deterministic policy gradient algorithms," in *International conference on machine learning*, pp. 387–395, Pmlr, 2014.

[14] P. N. Kolm and G. Ritter, "Dynamic replication and hedging: A reinforcement learning approach," *The Journal of Financial Data Science*, vol. 1, no. 1, pp. 159–171, 2019.