Deep hedging of Multi-Option Portfolios Using Real Market Data

CAPSTONE PROJECT REPORT

Siddhanth Sudhir Rau
Student ID: 1020221100

Supervisor:
Prof. Sandeep Juneja

Department of Computer Science
Ashoka University

27 December 2025

ACKNOWLEDGEMENT

I would like to express my sincere gratitude to my supervisor, **Prof. Sandeep Juneja**, for their continuous guidance, encouragement, and insightful feedback throughout the duration of this capstone project. Their expertise in Quantitative Finance and Machine Learning has been invaluable in shaping the direction and rigor of this work.

I am also deeply thankful to my faculty and peers for fostering an intellectually stimulating environment.

CERTIFICATE

This is to certify that the capstone project report titled: **"Deep Hedging of Multi-Option Portfolios Using Real Market Data"** submitted by **Siddhanth Sudhir Rau** (ID: **1020221100**) to the **Department of Computer Science**, **Ashoka University**, is a record of bona fide work carried out by the student under my supervision and guidance.

This work is submitted in partial fulfillment of the requirements for the award of the degree of **B.Sc. Computer Science**. To the best of my knowledge, the results embodied in this report have not been submitted to any other University or Institute for the award of any degree or diploma.

The work has been found to be satisfactory and meets the standards required for this capstone project.

Supervisor:

(Signature) ________________________________
Prof. Sandeep Juneja
Department of Computer Science
Ashoka University
Date: _____________________

Contents

1 INTRODUCTION

1.1 Overview

Modern financial markets are characterized by high-frequency fluctuations, complex derivative products, and non-linear risk profiles. Traditional hedging techniques, primarily rooted in the classical Black–Scholes–Merton (BSM) framework, rely on strong assumptions such as continuous-time trading, frictionless markets, and dynamically complete asset universes. However, real markets violate these assumptions: liquidity frictions, jumps, transaction costs, and incomplete markets introduce hedging errors that cannot be captured by classical models.

In recent years, deep learning has emerged as a powerful tool capable of approximating highly non-linear functionals and optimal control policies. The seminal work "Deep Hedging" (Buehler et al., 2019) formulated hedging as a stochastic optimal control problem and introduced a deep neural network–based framework capable of learning hedging strategies directly from simulated or historical data, without relying on closed-form models for dynamics.

This capstone project implements, extends, and adapts the deep hedging framework to a realistic, market-facing problem: **Hedging a portfolio of real AAPL option contracts using a data-driven deep neural hedger trained on historical returns and bootstrapped simulated price paths.**

Unlike the original paper—which primarily relies on simulated market scenarios (e.g., Heston, local volatility, or market-simulation engines)—this work incorporates real daily AAPL closing prices and real listed AAPL option chains into both training and evaluation. Consequently, the project bridges the gap between theoretical research and practical financial engineering.

1.2 Objective of the Study

The central objective of this capstone project is to design, implement, and evaluate a deep hedging system capable of managing the risks of a multi-option portfolio using real market data. This includes the following goals:

1. Reproduce and formalize the deep hedging mathematical framework as presented in Buehler et al. (2019).
2. Construct a multi-option portfolio using real AAPL option chain data.
3. Generate historical bootstrapped price paths for training the neural hedger.
4. Train two neural networks:
   (a) $\phi(0)$: hedger with no liability (benchmark risk measure).
   (b) $\phi(Z)$: hedger managing the option payoff liability.
5. Estimate the indifference price, defined as:

$$p(Z) = \rho(X) - \rho(X - Z) \tag{1}$$

   for coherent and convex risk measures (CVaR / entropic risk).
6. Conduct a complete monetary PnL simulation, including:
   (a) Premium intake.
   (b) Transaction costs.
   (c) Hedging costs.
   (d) Portfolio value path.

   (e) Final cash distribution.

7. Compare results against:
   (a) A naked short strategy.
   (b) A Black–Scholes delta hedge baseline.
8. Evaluate the learned strategy on an unseen real future price path.

These objectives collectively form a rigorous end-to-end study of deep hedging in a real-world setting.

1.3 Significance of the Study

This project is academically and practically significant for the following reasons:

1. **Bridging Theory and Practical Markets:** While the original deep hedging paper uses simulated paths, this work applies the framework to real AAPL prices and options, demonstrating practical viability.
2. **Improving Hedging in Incomplete Markets:** Markets with jumps, stochastic volatility, liquidity constraints, and transaction costs cannot be fully hedged linearly. Deep hedging provides a flexible alternative.
3. **Learning Optimal Nonlinear Policies:** Neural networks approximate optimal hedging strategies that are non-linear in time and state variables—beyond what delta hedging can capture.
4. **Risk Measure Integration:** Incorporating CVaR and entropic risk enables consistent risk-averse learning, aligning with regulatory and institutional constraints.
5. **Data-Driven Price Path Generation:** Bootstrap resampling from real returns enables realistic scenario generation while remaining model-free.

1.4 Research Questions

The study aims to answer the following research questions:

1. Can neural networks learn effective hedging strategies directly from historical market data?
2. How do learned strategies compare to classical hedging methods such as Black–Scholes delta hedging or no hedge?
3. What is the effect of different risk measures (CVaR vs. entropic) on:
   (a) Tail behavior of terminal PnLs.
   (b) Hedging aggressiveness.
   (c) Optimal trading frequency.
   (d) Hedging costs.
4. Can the deep hedger generalize to unseen real market price paths?

2 BACKGROUND AND THEORETICAL FRAMEWORK

2.1 Introduction

The challenge of pricing and hedging derivative portfolios in real markets arises from the fundamental fact that financial markets are incomplete and subject to frictions. Classical models such as Black–Scholes, Dupire local volatility, or stochastic volatility models assume frictionless markets and continuous trading, implying the existence of an equivalent martingale measure under which hedging is perfectly replicating. Under these assumptions, hedging reduces to computing sensitivities (Greeks) and pricing is linear in the derivative payoff.

In practice, none of these assumptions hold. Trading incurs transaction costs, bid/ask spreads, illiquidity, market impact, and risk limits. Furthermore, institutional traders operate under capital constraints and cannot hedge perfectly. As the Deep Hedging paper emphasizes, real-world hedging is performed in a non-linear environment, where pricing and risk depend on the current book composition, liquidity, and risk appetite.

The Deep Hedging framework seeks to reinterpret hedging as a stochastic optimal control problem on a discrete-time, frictional market, where trading strategies are parametrized by deep neural networks. This framework is rigorous, model-agnostic, and grounded in convex risk measures, allowing hedges to be optimized under realistic constraints and trading costs, without relying on risk-neutral pricing or Greeks.

This chapter provides a detailed account of the background theory necessary to understand and implement deep hedging, including the market model with frictions, trading strategies, convex risk measures, and the fundamental concept of indifference pricing. All mathematical definitions, processes, and equations follow the formalism presented in the Deep Hedging paper and are adapted to the setting of this capstone project.

2.2 Discrete-Time Financial Market with Frictions

The Deep Hedging framework operates on a discrete-time market, which is both realistic and computationally tractable. Let the trading dates be:

$$0 = t_0 < t_1 < \cdots < t_n = T, \tag{2}$$

where T is the final horizon, typically the maximum maturity of the hedged portfolio.

Probability Space and Information Structure

This is an finite probability space:

$$\Omega = \{\omega_1, \omega_2, \ldots, \omega_N\}, \quad P[\{\omega_i\}] > 0, \tag{3}$$

as described in the paper *Deep Hedging*. Random variables are elements of:

$$\mathscr{X} := \{X : \Omega \to \mathbb{R}\}. \tag{4}$$

At each time t_k, there is observable market information I_k, valued in $\mathbb{R}^r$. These may include:

1. Mid-prices of instruments.
2. Implied volatilities.
3. Liquidity metrics.

4. News signals.
5. Balance-sheet information.
6. Past hedges or features generated by the trader.

The information filtration is generated by:

$$\mathscr{F}_k = \sigma(I_0, \ldots, I_k). \tag{5}$$

Thus, any decision made at time t_k can use all available information up to that point. As the paper highlights: "This is the richest available feature set for any decision taken at time t_k". This interpretation is central: a hedging neural network may use any feature that is $\mathscr{F}_k$-measurable.

2.3 Hedging Instruments and Price Processes

The market contains d tradable hedging instruments, whose mid-prices are given by an $\mathbb{R}^d$-valued, adapted process:

$$S = (S_k)_{k=0,\ldots,n}. \tag{6}$$

No assumption is made about the existence of an equivalent martingale measure, or about the structure of S. It may include:

1. The underlying stock.
2. Liquid vanilla options.
3. Variance swaps.
4. Futures.
5. Any liquid instrument.

Crucially, instruments may not be tradable at all times (e.g., an option not yet listed). These liquidity constraints are included naturally in the friction model.

Liability Portfolio

Let $Z \in \mathscr{F}_T$ denote the liability payoff, representing the derivative portfolio being hedged. The payoff can be:

1. A single option.
2. A multi-option portfolio.
3. A mixture of OTC and listed products.

No model-based pricing, Greeks, or martingale representation is required. The only input is the payoff $Z(\omega)$ under each scenario.

2.4 Trading Strategies under Frictions

A hedging strategy is a predictable, $\mathscr{F}_{k-1}$-measurable process:

$$\delta_k = (\delta_k^1, \ldots, \delta_k^d), \tag{7}$$

where δ_k^i is the number of units of instrument i held between t_k and t_{k+1}.

Trades incur frictions, modeled as a cost function:

$$c_k(n_k) : \mathbb{R}^d \to \mathbb{R}, \tag{8}$$

where $n_k = \delta_k - \delta_{k-1}$ is the trade size.

The paper gives typical examples such as:

1. Proportional transaction costs:
$$c_k(n) = \sum_{i=1}^{d} \varepsilon |n_i| S_k^i. \tag{9}$$
2. Liquidity limitations.
3. Market-impact–based costs.

The accumulated cash from hedging evolves as:

$$X_T^{\delta} = -Z + \sum_{k=0}^{n-1} \delta_k \cdot (S_{k+1} - S_k) - \sum_{k=0}^{n} c_k(\delta_k - \delta_{k-1}). \tag{10}$$

This is the terminal hedging PnL under strategy δ.

2.5 Convex Risk Measures

Central to the deep hedging framework is the use of convex risk measures to evaluate terminal cashflows in incomplete markets. A convex risk measure is a functional $\rho : \mathscr{X} \to \mathbb{R}$, satisfying:

1. **Monotonicity:**
$$X \leq Y \implies \rho(X) \geq \rho(Y). \tag{11}$$
2. **Convexity:**
$$\rho(\lambda X + (1-\lambda)Y) \leq \lambda \rho(X) + (1-\lambda)\rho(Y), \quad \lambda \in [0,1]. \tag{12}$$
3. **Translation Invariance:**
$$\rho(X+m) = \rho(X) - m, \quad m \in \mathbb{R}. \tag{13}$$
4. **Cash Subadditivity** (or Cash-Additivity depending on convention).

The defining interpretation is that $\rho(X)$ is the minimal capital required to make the risky loss profile X acceptable.

Dual Representation

If the risk measure is lower semicontinuous, the paper notes it often admits a dual representation of the form:

$$\rho(X) = \sup_{Q \in \mathscr{Q}} \left(E_Q[-X] - \alpha(Q) \right), \tag{3.1}$$

where:

1. Q is a set of probability measures.
2. $\alpha(Q)$ is a penalty for choosing measure Q.

This representation shows the tight connection between risk measures and robust expectations.

The Entropic Risk Measure

An important special case, used heavily in deep hedging numerical examples, is the entropic risk measure:

$$\rho_\lambda(X) = \frac{1}{\lambda} \log E[e^{-\lambda X}], \quad \lambda > 0. \tag{3.4}$$

This arises naturally from exponential utility optimization and has many desirable properties:

1. Differentiable.

2. Strictly convex.
3. Sensitive to tail risks.
4. Smoothly interpolates between risk-neutral ($\lambda \to 0$) and highly risk-averse ($\lambda \to \infty$) regimes.

Average Value-at-Risk (CVaR)

The paper defines CVaR as:

$$\rho(X) = \frac{1}{1-\alpha} \int_0^{1-\alpha} \mathrm{VaR}_\gamma(X)\, d\gamma, \tag{5.1}$$

where:

$$\mathrm{VaR}_\gamma(X) = \inf\{m : P(X < -m) \leq \gamma\}. \tag{14}$$

CVaR is coherent, convex, and widely used in risk regulation.

2.6 Indifference Pricing

Indifference pricing is central to hedging with convex risk measures. The buyer's indifference price of a payoff Z is defined as:

$$p(Z) = \rho(X) - \rho(X - Z). \tag{3.2}$$

In deep hedging:

1. X is the terminal cashflow from hedging a zero payoff ("neutral strategy").
2. $X - Z$ is the terminal cashflow after liability Z is included.

Thus the indifference price is the premium that equalizes the risk of holding the liability and not holding it. The Deep Hedging algorithm computes this quantity numerically by training two neural networks:

1. $\phi(0)$: hedge with no liability.
2. $\phi(Z)$: hedge with the liability.

Then:

$$p(Z) = \rho(X^{\phi(0)}) - \rho(X^{\phi(Z)}). \tag{15}$$

This definition requires no model assumptions, no martingale measures, and integrates transaction costs seamlessly.

2.7 Deep Hedging as an Optimal Control Problem

Given the terminal hedging PnL X_T^δ, the hedging problem reduces to:

$$\min_{\delta \in \mathscr{A}} \rho(X_T^\delta), \tag{3.3}$$

where $\mathscr{A}$ is the set of feasible strategies.

The Deep Hedging paper proves (Section 4) that neural networks can approximate any optimal strategy to desired accuracy, due to universal approximation results. In practice:

1. Neural networks parametrize the strategy:

$$\delta_k = \phi_\theta(I_0, \ldots, I_k). \tag{16}$$

2. Gradient-based optimizers minimize the empirical risk.

3. The market model enters only via simulated or historical scenarios.

The optimization is fully model-agnostic.

3 LITERATURE REVIEW AND GAP ANALYSIS

3.1 Introduction

The problem of hedging contingent claims in financial markets has been studied extensively in mathematical finance for more than five decades. Traditional approaches rely on assumptions that guarantee market completeness, continuous trading, and frictionless execution. In such settings, derivative prices are uniquely determined through arbitrage arguments, and hedging can be achieved through exact replication. However, real markets violate nearly all ideal assumptions: traders face discrete trading opportunities, transaction costs, illiquidity, stochastic volatility, market impact, and model uncertainty. Moreover, the hedging of complex portfolios often occurs under deep nonlinearities induced by capital constraints and risk limits.

Recent advancements in machine learning, particularly deep reinforcement learning and deep neural networks, provide new tools for approximating optimal hedging strategies under realistic market constraints. The *Deep Hedging* framework (Buehler et al., 2019) represents one of the most significant applications of deep learning to quantitative finance, demonstrating that complex hedging strategies can be learned end-to-end without explicit modeling of risk-neutral dynamics.

This chapter situates deep hedging within the broader literature on pricing and hedging, including classical replication methods, utility-based optimization, convex risk measures, and machine learning–driven stochastic control. The chapter concludes with a detailed gap analysis, identifying why traditional methods are insufficient and how deep hedging fills a critical methodological gap.

3.2 Classical Hedging Theory in Complete Markets

3.2.1 Arbitrage Pricing and Replication

The foundations of modern derivative pricing were established by Black, Scholes, and Merton (1973), who derived the pioneering formula for European option prices under geometric Brownian motion. The key insight is the construction of a self-financing replicating strategy:

$$dX_t = \delta_t \, dS_t, \tag{17}$$

such that the terminal wealth replicates the derivative payoff Z. Under the assumption of no-arbitrage, a unique risk-neutral measure $\mathbb{Q}$ exists such that the price is given by the discounted expectation:

$$\pi(Z) = \mathbb{E}_{\mathbb{Q}}[e^{-rT} Z]. \tag{18}$$

This theory is elegant but relies critically on market completeness and frictionless trading—assumptions that fail in practice.

3.2.2 Delta Hedging and Greeks

Given the price function $V(t,S)$, the classical delta hedge is derived via Ito's Lemma:

$$\delta_t = \frac{\partial V}{\partial S}(t, S_t). \tag{19}$$

However, these quantities lose practical relevance when trading is discrete, volatility is stochastic, jumps occur, markets exhibit price impact, or transaction costs distort pricing. In such realistic scenarios, classical delta hedging becomes merely an approximation—often a poor one for multi-asset or exotic portfolios.

3.3 Hedging in Incomplete Markets

3.3.1 Quadratic Hedging and Mean–Variance Approaches

When markets are incomplete, perfect replication is impossible. Mean–variance hedging relaxes the replication constraint by minimizing the squared hedging error:

$$\min_{\delta} \mathbb{E}[(Z - X_T^{\delta})^2]. \tag{20}$$

This framework is mathematically tractable but fails to incorporate transaction costs or nonlinear risk constraints. Moreover, quadratic loss functions penalize gains and losses symmetrically and underestimate tail risk, making the approach undesirable for institutional traders who are primarily concerned with downside protection.

3.3.2 Utility-Based Hedging

Utility indifference pricing is based on maximizing expected utility:

$$\sup_{\delta} \mathbb{E}[U(X_T^{\delta})]. \tag{21}$$

For the specific case of exponential utility $U(x) = -e^{-\lambda x}$, indifference pricing leads to the following pricing formula:

$$p(Z) = \frac{1}{\lambda} \log \left(\frac{\sup_{\delta} \mathbb{E}[e^{-\lambda X_T^{\delta}}]}{\sup_{\delta} \mathbb{E}[e^{-\lambda (X_T^{\delta} - Z)}]} \right). \tag{22}$$

This formulation is mathematically equivalent to optimizing entropic risk measures. However, solving Equation (22) analytically becomes computationally intractable for large portfolios and typically requires an explicit probabilistic model, which is often misspecified in practice.

3.4 Convex Risk Measures and Modern Risk-Based Hedging

Convex risk measures map a random variable $X \to \rho(X)$ and satisfy axioms of monotonicity, convexity, and translation invariance. Their dual representation is given by:

$$\rho(X) = \sup_{\mathbb{Q}} \left(\mathbb{E}_{\mathbb{Q}}[-X] - \alpha(\mathbb{Q}) \right). \tag{23}$$

Two primary risk measures used in modern quantitative finance are:

1. **Entropic Risk Measure:**
$$\rho_{\lambda}(X) = \frac{1}{\lambda} \log \mathbb{E}[e^{-\lambda X}]. \tag{24}$$

2. **Conditional Value-at-Risk (CVaR):**
$$\mathrm{CVaR}_{\alpha}(X) = \frac{1}{1-\alpha} \int_{\alpha}^{1} \mathrm{VaR}_u(X)\, du. \tag{25}$$

While theoretically robust, computing optimal hedges under these risk measures using traditional numerical methods (such as grid-based PDEs) is numerically difficult due to nonlinearities and the curse of dimensionality.

3.5 Frictions, Transaction Costs, and Market Impact

3.5.1 Transaction Cost Models

Hedging with proportional costs invalidates classical replication arguments. Nonlinear PDEs and market impact models have been developed to address this, utilizing general cost functions of the form:

$$c_k(n_k) = \lambda_k |n_k|^\beta S_k, \tag{26}$$

with $\beta = 1$ representing proportional costs. However, analytical solutions exist only for very specific, simplified cases.

3.5.2 Liquidity and Price Impact

Price impact and liquidity considerations strongly motivate hedging frameworks that work directly with discrete trading and frictions. Traditional continuous-time models often fail to capture the discrete nature of liquidity provision, necessitating the discrete-time approach adopted in Deep Hedging.

3.6 Machine Learning Approaches to Hedging

3.6.1 Reinforcement Learning (RL)

Reinforcement Learning strategies have been proposed for trading and execution. While promising, RL approaches in finance often suffer from a lack of convergence guarantees, overfitting to training data, and difficulty in explicitly incorporating coherent risk measures into the reward function.

3.6.2 Deep Learning for Derivative Pricing

Deep learning has been widely used to approximate PDE solutions and pricing operators. However, the majority of these works focus on accelerating the calculation of prices within existing models (e.g., accelerating Heston model calibration) rather than solving the hedging problem under frictions or convex risk measures.

3.7 Deep Hedging and the Modern Shift

3.7.1 The Deep Hedging Framework

Deep Hedging treats hedging as a sequential learning problem using neural networks to parametrize trading strategies:

$$\delta_k = \phi_\theta(I_0, \dots, I_k). \tag{27}$$

The network is trained to minimize a convex risk measure over the terminal PnL:

$$\min_\theta \rho(X_T^{\phi_\theta}). \tag{28}$$

This approach offers several distinct advantages over classical methods:

1. **Model-Free Dynamics:** It does not require a specific probabilistic model of S_t; it can learn directly from data.

2. **Frictions Integration:** It naturally incorporates transaction costs, liquidity constraints, and market impact into the loss function.
3. **Scalability:** It handles large derivative portfolios efficiently using GPU acceleration.
4. **Risk-Awareness:** It optimizes directly under institutional risk measures (CVaR/Entropic) rather than proxy objectives like variance.

3.7.2 Gap Analysis and Contribution

Despite the theoretical success of Deep Hedging, existing literature predominantly relies on simulated market environments (e.g., Heston or Local Volatility simulations) to train and test these networks. There is a significant gap in the application of this framework to **real, empirical market data**.

Identified Gaps:

1. **Data Realism:** Most studies assume the underlying process is known (simulation). Real markets exhibit jumps and non-stationarity not captured by standard simulations.
2. **Practical Implementation:** There is limited literature on the end-to-end implementation of deep hedging pipelines using real option chain data.

Contribution of this Study: This capstone project addresses these gaps by implementing the Deep Hedging framework using **real AAPL market data**. By training on bootstrapped historical paths and hedging actual option contracts, this study evaluates the viability of Deep Hedging in a realistic, model-agnostic setting, moving beyond theoretical simulations to practical application.

4 MATHEMATICAL FRAMEWORK AND PROBLEM FORMULATION

4.1 Overview

This chapter formalizes the problem that this project aims to solve: how to hedge a portfolio of options written on AAPL using a deep neural network, in the presence of transaction costs and market incompleteness.

Instead of searching for a perfect replication strategy (which is usually impossible in practice), I adopted the *Deep Hedging* perspective (Buehler et al., 2019): I treated hedging as an optimization problem where I chose a trading strategy that minimized a risk measure of the final profit-and-loss (PnL). The optimization is done over a parametric family of strategies represented by a neural network.

I now describe:

1. The discrete-time market model.
2. How the hedging PnL is computed.
3. How strategies are parametrized by a neural network.
4. Which risk measures is used.
5. How indifference pricing is obtained from the risk measure.

4.2 Discrete-Time Market Model

4.2.1 Time Grid and Price Paths

Discrete time, on trading dates are used:

$$0 = t_0 < t_1 < \cdots < t_{n-1} < t_n = T, \tag{29}$$

where T is the final horizon (chosen at least as large as the maximum option maturity in the portfolio).

The underlying asset is AAPL. Its (discounted) price process is denoted:

$$S_0, S_1, \ldots, S_n, \tag{30}$$

where S_k is the price at time t_k.

In the implementation, these paths are obtained by bootstrap resampling of historical log-returns:

1. I took a long time series of AAPL daily prices.
2. I computed log-returns $R_t = \log(S_{t+1}/S_t)$.
3. I then generated synthetic paths by sampling blocks of returns and compounding them, starting from a recent spot price.

Mathematically, each path corresponds to a scenario ω,

$$S_k = S_k(\omega), \quad k = 0, \ldots, n. \tag{31}$$

I do not assume any particular stochastic differential equation or parametric model (like Black–Scholes or Heston); all dynamics come from the empirical return distribution.

4.2.2 Option Portfolio (Liability)

The liability which has to be hedged is a portfolio Z of European options on AAPL. Suppose there are m individual contracts, each with:

1. Strike K_j.
2. Maturity date (mapped to time index) $\tau_j \in \{0, \ldots, n\}$.
3. Type (call/put).
4. Quantity q_j (positive if short the option, i.e., sell it).

The payoff of option j at its maturity is:

$$\text{Call: } \Pi_j(S_{\tau_j}) = \max(S_{\tau_j} - K_j, 0), \tag{32}$$

$$\text{Put: } \Pi_j(S_{\tau_j}) = \max(K_j - S_{\tau_j}, 0). \tag{33}$$

The total liability payoff at the terminal time horizon is then:

$$Z = \sum_{j=1}^{m} q_j \Pi_j(S_{\tau_j}). \tag{34}$$

Here "liability" is from the perspective of someone who has sold these options and must pay out their intrinsic value at maturity.

4.3 Trading Strategy and PnL

4.3.1 Positions and Trades

A trading strategy is a sequence of positions in the hedging asset (AAPL):

$$\delta_0, \delta_1, \ldots, \delta_{n-1}, \tag{35}$$

where δ_k is the number of shares held between times t_k and t_{k+1}.

I then set $\delta_{-1} := 0$ (no initial inventory). The trade at time t_k is:

$$n_k = \delta_k - \delta_{k-1}. \tag{36}$$

If $n_k > 0$, then buy shares; if $n_k < 0$, sell shares.

4.3.2 Transaction Costs

Trading is not free. Proportional transaction costs are incorporated of the form:

$$c_k(n_k) = \varepsilon\, |n_k|\, S_k, \tag{37}$$

where $\varepsilon > 0$ is a small cost parameter (e.g. 0.1% per trade). This term penalizes frequent, large rebalancing. This form is simple but realistic enough to capture the trade-off between hedging performance and cost.

4.3.3 Cash Account and Self-Financing Condition

Let X_k be the cash in the hedging account at time t_k. Starting with some initial cash X_0, the cash evolves according to:

1. When the position is changed from δ_{k-1} to δ_k, transaction cost is paid.
2. Between t_k and t_{k+1}, the position δ_k is exposed to the price change $S_{k+1} - S_k$.

A simple way to encode this is:

$$X_{k+1} = X_k - \delta_k(S_{k+1} - S_k) - c_k(n_k), \quad k = 0, \ldots, n-1. \tag{38}$$

Interpretation:

1. $-\delta_k(S_{k+1} - S_k)$: the trading gain or loss due to holding δ_k over $[t_k, t_{k+1}]$.
2. $-c_k(n_k)$: the transaction cost of changing the position at time t_k.

At the final time $t_n = T$, typically liquidate the position by trading from δ_{n-1} back to zero, generating one more cost term: $c_n(n_n)$, with $n_n = -\delta_{n-1}$.

4.3.4 Terminal Wealth and PnL

Let X_T be the cash after all trading and transaction costs, before paying the option payoff. The terminal wealth after paying the liability is:

$$W_T = X_T - Z. \tag{39}$$

To measure performance relative to a natural benchmark, I defined a baseline B, usually:

$$B = X_0 + (\text{total option premiums received at time } t_0). \tag{40}$$

The hedging profit-and-loss (PnL) relative to that baseline is:

$$\Pi = W_T - B. \tag{41}$$

If $\Pi > 0$: the model did better than just collecting premiums and holding cash. If $\Pi < 0$: our hedging (plus liability) lost money relative to that baseline.

The entire deep hedging problem can now be viewed as: *Choose a sequence of positions* $\delta_0, \dots, \delta_{n-1}$ *(based on the observed price path) to make the distribution of* Π *as favorable as possible under a chosen risk measure.*

4.4 Neural Network Parametrization of the Hedging Strategy

The strategy δ_k at time t_k must be based only on information available at that time. In practice, the model does not directly work with the filtration; instead, build a feature vector.

In this project, at time t_k, features are used:

$$F_k = (\log S_k,\, \tau_k,\, \delta_{k-1}), \tag{42}$$

where:

1. $\log S_k$ is the current log-price.
2. τ_k is a normalized time-to-horizon (e.g., $(n-k)/n$).
3. δ_{k-1} is the current position (so the network knows inventory).

Then neural network ϕ_θ maps features to the next position:

$$\delta_k = \phi_\theta(F_k). \tag{43}$$

Here:

1. θ are the learnable parameters (weights and biases).
2. ϕ_θ is typically a small multilayer perceptron (MLP) with nonlinear activation functions and layer normalization.

This gives us a family of strategies indexed by θ. Our problem becomes: *Find* θ *such that the induced PnL* $\Pi(\theta)$ *has minimal risk according to our chosen risk measure.*

4.5 Risk Measures: Entropic and CVaR

To evaluate the quality of a hedging strategy, the risk measures rather than just expected PnL is used.

4.5.1 Entropic Risk Measure

For a random PnL X, the entropic risk measure with risk aversion parameter $\lambda > 0$ is defined as:

$$\rho_\lambda(X) = \frac{1}{\lambda} \log\left(E[e^{-\lambda X}]\right). \tag{44}$$

Interpretation:

1. When λ is small, this is close to $\rho_\lambda(X) \approx -E[X] + \frac{\lambda}{2}\mathrm{Var}(X)$, so it looks like a mean–variance objective.
2. Larger λ makes the measure more sensitive to large negative outcomes (tail risk), so the learned hedge tends to be more conservative.

In practice, given simulated PnLs $\{X^{(i)}\}_{i=1}^{N}$ estimate:

$$\hat{\rho}_\lambda(X) = \frac{1}{\lambda} \log \left(\frac{1}{N} \sum_{i=1}^{N} e^{-\lambda X^{(i)}} \right). \tag{45}$$

To avoid numerical overflow, clip the exponent $-\lambda X^{(i)}$ to a reasonable range before applying the exponential.

4.5.2 Conditional Value-at-Risk (CVaR)

Let X again be the PnL and define the loss $L = -X$ (so large negative PnL corresponds to large positive loss). Given a confidence level $\alpha \in (0,1)$, the Value-at-Risk (VaR) at level α is essentially the α-quantile of the loss distribution.

The Conditional Value-at-Risk (CVaR) at level α is the expected loss that occurs in the worst $(1-\alpha)$ fraction of scenarios:

$$\mathrm{CVaR}_\alpha(X) = E[L \mid L \geq \mathrm{VaR}_\alpha(X)]. \tag{46}$$

So, for $\alpha = 0.95$, CVaR measures the average loss in the worst 5% of cases.

Empirically, if there are samples $\{X^{(i)}\}$:

1. Compute losses $L^{(i)} = -X^{(i)}$.
2. Find the empirical 95% quantile of $L^{(i)}$, call it $q_{0.95}$.
3. Compute:

$$\widehat{\mathrm{CVaR}}_{0.95}(X) = \frac{1}{\#\{i : L^{(i)} \geq q_{0.95}\}} \sum_{i:L^{(i)} \geq q_{0.95}} L^{(i)}. \tag{47}$$

Minimizing CVaR focuses the optimization on tail risk, which is particularly important for risk management and regulatory reasons.

4.6 Optimization Problem: Deep Hedging Objective

Given the previous definitions, the Deep Hedging problem in this project can be stated as:

1. Strategies of the form $\delta_k = \phi_\theta(F_k)$ are considered.
2. For each θ, compute the induced PnL $\Pi(\theta)$ by simulating many price paths and applying Equations (38)–(41).
3. Evaluate the risk of $\Pi(\theta)$ using either the entropic risk measure (44) or CVaR (46).

The learning objective is:

$$\min_\theta \rho(\Pi(\theta)), \tag{48}$$

where ρ is either ρ_λ (entropic) or CVaR_α.

In practice, replace the theoretical expectation by a Monte Carlo estimate over a large simulated dataset of paths, and minimize the empirical risk via gradient-based training (Adam optimizer in TensorFlow).

4.7 Indifference Pricing

One of the key conceptual advantages of Deep Hedging is that it naturally yields a risk-based price for the liability portfolio, called the indifference price.

The idea is:

1. First, consider a trader with no liability, who just manages their position optimally to minimize risk. Let the associated minimal risk be:

$$\rho_0^\star = \min_\theta \rho(\Pi_0(\theta)), \tag{49}$$

where $\Pi_0(\theta)$ is the PnL when no option payoff is present ($Z = 0$).
2. Now, suppose the trader sells the portfolio Z at price p at time t_0 and then hedges optimally. The PnL becomes $\Pi_Z(\theta) + p$ (add the premium p to the terminal cash). The minimal risk in this case is:

$$\rho_Z^\star(p) = \min_\theta \rho(\Pi_Z(\theta) + p). \tag{50}$$

The indifference price p_{ind} is defined as the price at which the trader is indifferent between having the liability and not having it, in terms of risk:

$$\rho_0^\star = \rho_Z^\star(p_{\text{ind}}). \tag{51}$$

Using the translation invariance of risk measures ($\rho(X+m) = \rho(X) - m$), simplify:

$$\rho_Z^\star(p) = \min_\theta \rho(\Pi_Z(\theta) + p) = \min_\theta(\rho(\Pi_Z(\theta)) - p) = \rho_Z^\star(0) - p. \tag{52}$$

Setting $\rho_0^\star = \rho_Z^\star(p_{\text{ind}})$ gives:

$$\rho_0^\star = \rho_Z^\star(0) - p_{\text{ind}}, \tag{53}$$

so

$$p_{\text{ind}} = \rho_Z^\star(0) - \rho_0^\star. \tag{54}$$

Interpretation: The indifference price is the difference between the minimal risk when hedging with liability and the minimal risk when hedging without liability.

In the implementation, approximate this as:

1. Train a “zero” hedger on $Z = 0$ and estimate $\rho_0^\star$.
2. Train a “portfolio” hedger on the real payoff Z and estimate $\rho_Z^\star(0)$.
3. Set:

$$\hat{p}_{\text{ind}} = \hat{\rho}_Z^\star(0) - \hat{\rho}_0^\star. \tag{55}$$

Bootstrap resampling of test PnLs is used to obtain confidence intervals for $\hat{p}_{\text{ind}}$.

5 SYSTEM DESIGN, SCOPE, AND METHODOLOGY

5.1 Introduction

This chapter presents a complete description of the computational architecture that implements the Deep Hedging framework for a multi-option AAPL portfolio. The system is built around a modular, extensible architecture in which each component—from market data processing to neural network training, risk-measure-based optimization, indifference pricing, and monetary simulation—is encapsulated in a dedicated class or function. This structure aligns with modern standards in both quantitative finance and applied machine learning, ensuring reproducibility, clarity, and robustness.

Related Code: This section introduces the entire module `deep_hedging_refactor_paper_aligned_fixed.py`.

5.2 Scope and Objectives

The implementation is designed to learn dynamic hedging strategies for a realistic AAPL options portfolio using a deep neural network. Instead of relying on assumed stochastic models, the system uses non-parametric historical bootstrapping to generate price scenarios. It incorporates practical trading frictions such as transaction costs, constrained rebalancing, and multi-expiry option cash flows. The system then trains two hedgers—one for the zero-liability case and one for the actual liability—and computes an indifference price from their risk profiles. Extensive evaluation across simulated and real scenarios ensures robustness.

Related Code:

- `run_full_pipeline_paper_aligned()` (main orchestrator)
- `Trainer` class
- `IndifferencePricer` class

5.3 Market Data Stage: Acquisition and Preprocessing

The data pipeline begins with the `MarketData` class, which retrieves historical AAPL prices from Yahoo Finance. The system prioritizes the Adjusted Close field for accurate dividend-adjusted pricing but falls back to safe alternatives when necessary. All retrieved prices are stored in NumPy arrays for efficient computation, and corresponding timestamps are retained for precise option maturity alignment.

A safe and validated log-return computation method ensures no invalid values corrupt the scenario generator. Log-returns are defined as:

$$R_t = \log(S_{t+1}) - \log(S_t). \tag{56}$$

These returns form the empirical distribution used for bootstrapping. Additionally, a "holdout window" of recent real prices is extracted to serve as an unseen evaluation path—an approach inspired by the methodology of the original Deep Hedging paper.

Related Code:

- Class: `MarketData`
- Method: `load_prices()`
- Method: `safe_log_returns()`
- Method: `build_holdout_and_pool()`

5.4 Non-Parametric Price Simulation via Bootstrapping

Synthetic price paths used for training and validation are generated through a non-parametric bootstrapping method implemented in the `Bootstrapper` class. Rather than fitting a diffusion process, the system samples blocks of historical log-returns and compounds them into paths. This preserves heavy tails, volatility clustering, and other empirical features of equity markets without imposing parametric assumptions.

Each path is generated as:

$$S_t^{(i)} = S_0 \exp\left(\sum_{j=1}^{t} R_{\eta_i+j-1}\right), \tag{57}$$

where η_i is a random starting index for the i-th path.

When option maturities exceed the sampled window, the system attaches extra bootstrapped segments to the paths. This flexible construction guarantees that all paths are sufficiently long for payoff computation.

Related Code:

- Class: `Bootstrapper`
- Method: `bootstrap_paths_vectorized()`
- Method: `extend_paths_by_returns()`

5.5 Construction of the Option Portfolio

Using real option chain data supplied by the user, the system constructs a realistic portfolio of AAPL options. Two constructors support training and scenario-specific evaluation. The method filters out illiquid options, aligns expiries with the trading horizon, computes business-day maturities, and converts each valid option into an `OptionSpec`.

The total portfolio liability is then:

$$Z(\omega) = \sum_{j=1}^{m} q_j \cdot \Pi_j(S_{\tau_j}(\omega)), \tag{58}$$

where Π_j is a European call or put payoff function.

Related Code:

- Function: `build_real_option_portfolio_for_training()`
- Function: `build_real_option_portfolio_from_df()`
- Data Structure: `OptionSpec`
- Function: `portfolio_payoff()`

5.6 Feature Engineering for Neural Hedging

Each path is transformed into a sequence of feature vectors that encode essential market information for decision-making. At each time step, the hedger observes:

1. The log-price, $\log S_t$.
2. A normalized time variable representing remaining horizon.
3. The inventory state, δ_{t-1}.

These features are constructed in a fully vectorized manner and are supplied to the neural network at every decision step. This preserves the Markov property and ensures the network has sufficient information to reason about both price dynamics and its own past actions.

Related Code:

- Function: `build_features()`

5.7 Neural Network Hedger Architecture

The `HedgerNet` class defines a robust yet lightweight multilayer perceptron that maps observed features to a position δ_t. The architecture includes:

1. Dense layers with ReLU activations.
2. Layer normalization for gradient stability.
3. Dropout for regularization.
4. A final linear output layer producing the share position.

This architecture balances expressiveness with generalization, avoiding overfitting while capturing non-linear structure in the hedging problem.

Related Code:

- Class: `HedgerNet`

5.8 Risk-Based Training Procedure

Training involves unfolding the hedging policy along simulated price paths and computing the resulting PnL. The training loop iteratively:

1. Predicts the position δ_t.
2. Computes the incremental trade $n_t = \delta_t - \delta_{t-1}$.
3. Accumulates transaction costs $c_t = \varepsilon|n_t|S_t$.
4. Updates PnL according to $\text{PnL} \leftarrow \text{PnL} + \delta_{t-1}(S_{t+1} - S_t)$.
5. Deducts option payoffs at their maturity.

The terminal PnL is then evaluated through a convex risk measure. The system supports:

- **CVaR:** $\text{CVaR}_\alpha(X) = \mathbb{E}[L \mid L \geq \text{VaR}_\alpha], \quad L = -X.$

- **Entropic Risk:** $\rho_\lambda(X) = \frac{1}{\lambda}\log\mathbb{E}[e^{-\lambda X}].$

Both risk measures are implemented in TensorFlow and are fully differentiable.

Two hedgers are trained:

1. ϕ_{θ_0} for the zero-liability case.
2. ϕ_{θ_Z} for the actual liability.

This structure is essential for computing the indifference price.

Related Code:

- Class: `Trainer`

- Method: `train_hedger()`

- Method: `forward_train()`

- Method: `cvar_tf()`

- Method: `entropic_tf()`

5.9 Indifference Pricing Mechanism

Indifference pricing quantifies the premium at which selling the option portfolio leaves the trader's risk unchanged. After training both the zero-liability hedger and the liability hedger, the system computes empirical risk on a large test set and evaluates:

$$p_{\text{ind}} = \rho(\Pi_0^{\theta_0}) - \rho(\Pi_Z^{\theta_Z}). \tag{59}$$

A bootstrap procedure estimates confidence intervals by repeatedly resampling the PnL distribution and recomputing the indifference price.

Related Code:

- Class: `IndifferencePricer`
- Method: `bootstrap_indifference()`

5.10 Monetary Simulation and Portfolio-Level Diagnostics

Beyond purely statistical evaluation, the system performs a realistic monetary simulation that reconstructs the cash account through time. This simulation mirrors institutional PnL accounting by tracking:

1. Accumulated trading costs.
2. Evolving hedging inventory.
3. Portfolio value fluctuations.
4. Option payoffs at varying maturities.
5. Final PnL relative to a premium-adjusted baseline.

This enables direct comparison between the learned deep hedger, classical Black–Scholes delta hedging, and the naked short position. Comprehensive diagnostic plots provide insights into hedge behavior, stress sensitivity, and tail-risk characteristics.

Related Code:

- Function: `simulate_portfolio_money_on_paths()`
- Function: `monetary_test_report()`
- Function: `monetary_cash_report()`

5.11 Real Historical Scenario Evaluation

A dedicated evaluation routine applies the trained Deep Hedger to actual AAPL historical price sequences from a year held out from training. The system rebuilds a portfolio corresponding to the evaluation date, projects the hedger's decisions along the real path, and computes realized PnL. By comparing this performance to classical benchmarks, the analysis reveals how the Deep Hedger behaves under real market dynamics.

Related Code:

- Function: `real_options_scenario_report()`

5.12 Integrated Pipeline

All components described above are executed in an integrated, orderly sequence by the master function `run_full_pipeline_paper_aligned()`. This orchestrator coordinates data ingestion, return bootstrapping, feature engineering, portfolio building, two-stage hedger training, risk-indifference pricing, extensive simulated testing, real historical evaluation, and the generation of all diagnostic reports. This end-to-end pipeline forms a reproducible experimental framework consistent with modern standards in quantitative finance.

Related Code:

- Function: `run_full_pipeline_paper_aligned()`

5.13 Summary

The system presented in this chapter integrates all elements of the Deep Hedging methodology into a coherent computational architecture. From empirical data acquisition to neural network optimization under convex risk measures, and from synthetic path simulation to real scenario evaluation, every step is implemented with scientific precision and practical relevance. The result is a full-featured Deep Hedging engine that mirrors academic formulations while remaining applicable to real-world trading scenarios.

6 WORK DONE: IMPLEMENTATION, CHALLENGES, AND MITIGATIONS

This chapter documents the practical implementation of the deep hedging framework and provides a critical evaluation of the engineering challenges encountered. The implementation required integrating multiple theoretical constructs—convex risk measures, stochastic optimal control, and neural functional approximators—into a functioning system trained on real financial data. As such, the work demanded both conceptual understanding and substantial engineering effort. This chapter narrates the workflow and contextualizes the system design within the constraints and complexities encountered.

6.1 Implementation Overview

The deep hedging framework was implemented as a fully integrated Python system, largely aligned with the structure described in Bühler et al. (2019). The completed pipeline encompassed several non-trivial components working in tandem: historical data ingestion, bootstrap-based scenario generation, real-world option portfolio construction, neural hedger training, risk-sensitive optimization, monetary backtesting, and out-of-sample validation.

6.1.1 Data Preparation and Market Environment Construction

A significant portion of the work involved constructing a realistic and sufficiently rich market environment for training and evaluation. Historical price data for Apple Inc. (AAPL) spanning 2015–2024 was obtained using the `MarketData` class. The dataset comprised more than 2,500 daily observations, which provided an adequate foundation for non-parametric sampling. However, because the hedging horizon was determined by the maximum maturity of the real options (22 trading days), much of the long historical period had to be compressed into short horizon paths.

To create the training environment, a holdout window from late 2024 was extracted for validation, and the remaining historical prices were processed into a “pool” used for bootstrapping. Log returns from this pool were then sampled in contiguous blocks to generate simulated price paths. This was accomplished with the `Bootstrapper` class, which produced 12,000 training paths and 5,000 test paths, each of length 23 (22 increments). The resulting paths resembled realistic historical patterns without assuming any parametric model such as Black–Scholes or Heston.

This approach—historical block bootstrapping instead of a parametric simulator—is closer to authentic market microstructure and thus preserves the empirical return distribution, including volatility clustering, heavy tails, and jumps.

6.1.2 Real Options Portfolio Construction

A major accomplishment of this project was constructing a real, tradeable options portfolio from market data. The `build_real_option_portfolio_for_training` function filtered options based on:

1. Ticker (AAPL).
2. Quote date (2025-07-16).
3. Minimum volume ($\geq$ 10 contracts).
4. Minimum maturity of 30 days (to ensure non-trivial hedging dynamics).

Three options were chosen after filtering to form the liability portfolio:

1. AAPL250815C00220000 — Strike 220, Premium 3.25.
2. AAPL250815C00210000 — Strike 210, Premium 7.50.
3. AAPL250815C00225000 — Strike 225, Premium 1.98.

These contracts provided a concise but meaningful multi-option hedging challenge. Each option included its real premium, and the hedger was trained under the assumption of being short one contract of each. The combined premium at initiation was approximately 12.73 USD.

Constructing a real portfolio rather than assuming synthetic strikes required navigating numerous data inconsistencies, such as missing bid/ask quotes and expired symbols. Premium estimation fallback logic was implemented, preferring "mark" price and falling back to mid bid–ask or last traded price when necessary.

6.1.3 Neural Hedger Training and Optimization

The core of the implementation involved training two neural networks:

1. $\phi(0)$ — a hedger optimizing the objective when no liability is present.
2. $\phi(Z)$ — a hedger optimizing under the actual option payoff liability.

Both networks shared the same architecture (`HedgerNet`), comprising dense layers, layer normalization, and dropout regularization. They received three inputs at each time step: log-price of the underlying, normalized time-to-maturity, and previous hedge position.

Support for Multiple Risk Measures: The system was engineered to support configurable risk objectives, allowing for comparative experiments between:

1. **CVaR:** Minimizing the tail loss at a 95% confidence level.
2. **Entropic Risk:** Minimizing exponential utility with varying risk-aversion parameters (λ). The implementation specifically tested $\lambda = 1.0$ (high risk aversion) and $\lambda = 0.1$ (low risk aversion) to observe the impact on hedging aggressiveness.

The training loop, implemented in `Trainer.forward_train`, propagated hedging decisions through simulated paths, accumulating mark-to-market PnL, transaction costs, and final liability payoffs.

6.1.4 Risk-Based Pricing and Performance Evaluation

After both hedgers were trained, the difference in their empirical risk measures estimated the indifference price:

$$p(Z) = \phi(Z) - \phi(0). \tag{60}$$

Recognizing that $\phi(Z)$ may be poorly calibrated due to optimization noise, a bootstrap procedure was implemented in the `IndifferencePricer` class. This technique resampled terminal PnLs to yield a robust mean estimate and confidence intervals, correcting for the high variance inherent in single-run estimates.

The hedger was subsequently tested in a realistic monetary simulation, tracking cash balances, positions, trading costs, and option payoff settlements. This validated the deep hedging concept against benchmarks like Naked Shorting and Black-Scholes delta hedging.

6.2 Challenges Encountered

Throughout the project, several challenges arose, spanning data limitations, modelling complexity, training instability, and computational constraints.

6.2.1 Unrealistic Indifference Price Estimates

The most critical problem encountered was the disconnect between the raw indifference price and the bootstrap-adjusted one. The raw estimates were often orders of magnitude larger than the total premium. This resulted from the $\phi(Z)$ hedger's difficulty in converging to a stable global minimum. Possible causes identified include:

1. CVaR optimization is extremely sensitive to worst-case paths and can disproportionately weight rare extreme losses.
2. The joint optimization of hedging decisions while absorbing short option liability makes $\phi(Z)$ inherently more unstable than $\phi(0)$.

6.2.2 Limited and Noisy Options Data

Another major challenge stemmed from the scarcity and inconsistency of free historical options data. The dataset required extensive cleaning, including fixing type inconsistencies, resolving missing mark prices, and filtering zero-volume entries. Furthermore, the restriction to a single quote date (2025-07-16) limited the diversity of training scenarios regarding implied volatility regimes.

6.2.3 Optimization Instability with High Risk Aversion

While CVaR optimization is known to be difficult, training under Entropic Risk with high aversion ($\lambda = 1.0$) proved equally challenging. In these configurations, the objective function became dominated by extreme loss events. This often caused the network to fall into degenerate local minima where it effectively ceased active hedging, behaving similarly to a naked short position. This highlighted the sensitivity of Deep Hedging to hyperparameter selection.

6.2.4 Constraints Due to Limited Computational Resources

The project involved computationally heavy processes including the generation of thousands of bootstrapped paths and repeated multi-epoch neural network training. Because training occurred on CPU, extensive hyperparameter sweeps and architecture exploration were constrained. This limited the ability to fully investigate stability-enhancing configurations.

6.3 Mitigation Strategies Implemented

Despite the challenges outlined above, several mitigation strategies were applied to stabilize training, improve data quality, and ensure meaningful evaluation.

6.3.1 Architectural Stabilization Techniques

To address training instability, multiple architectural safeguards were incorporated:

1. Layer normalization standardized activations across layers.
2. Dropout (0.08) reduced overfitting and smoothed training.
3. Lower learning rates (3×10^{-4}) moderated gradient volatility.
4. Global norm gradient clipping prevented exploding gradients.
5. Controlled input feature design (log-price, time-to-maturity, previous hedge position) avoided unnecessary model complexity.

These choices collectively contributed to reliable convergence of $\phi(0)$ and partial stabilization of $\phi(Z)$.

6.3.2 Robust Monetary Simulation Framework

A comprehensive monetary accounting system was created to evaluate hedger performance under realistic trading frictions. This included:

1. Sequential position updates.
2. Cash-account tracking.
3. Transaction cost deduction.
4. Payoff settlement timing.
5. Final liquidation of positions.

This allowed fair comparison between the learned hedger, Black–Scholes delta hedge, and naked short position. The simulation framework provided meaningful and interpretable performance metrics, even when training outcomes were imperfect.

6.3.3 Bootstrap-Based Calibration of Indifference Pricing

Recognizing the unreliability of the raw $p(Z)$ estimate, a bootstrap-based approach was adopted for robust pricing. This method effectively averaged away random variations in $\phi(Z)$ and $\phi(0)$, producing a stable estimate and highlighting that the pricing issue stemmed from noisy hedger estimates, not conceptual flaws.

6.3.4 Enhanced Option Data Cleaning and Validation

To extract usable contracts from noisy real-world data, several mechanisms were implemented:

1. Handling non-numeric entries.
2. Fallback premium selection hierarchy.
3. Filtering low-volume contracts.
4. Business-day maturity calculation.
5. Validation that maturity does not exceed the simulated horizon.

Without these steps, the portfolio construction process would have repeatedly failed or produced unrealistic contracts unsuitable for hedging.

6.3.5 Organized Research and Interpretation Workflow

To overcome the conceptual complexity of the Deep Hedging paper, a structured reading and interpretation strategy was required. Core concepts such as dynamic risk measures, entropic pricing, and backward risk recursion were broken into smaller subsections and aligned directly with code components. This reduced abstraction gap and allowed faithful reproduction of the methodology.

7 RESULTS AND DISCUSSION

This section presents the empirical performance of the deep hedging framework implemented on a portfolio of three short AAPL call options. Results are reported for:

1. A primary CVaR-based training and testing run on bootstrapped simulated paths.
2. A real single-path scenario from the AAPL time series.
3. A controlled experiment comparing a CVaR objective with entropic risk measures at two different risk-aversion levels, $\lambda = 1.0$ and $\lambda = 0.1$.

In all cases, the underlying is AAPL, the hedging horizon is approximately 22 trading days (matching option maturity), and the portfolio consists of three short call options with strikes $K \in \{210, 220, 225\}$, quoted on 2025-07-16.

7.1 Experimental Setup Summary

Training and testing follow the same high-level pipeline. Historical AAPL prices were downloaded from January 2015 to December 2024 for training, and from January 2025 onwards for testing. Log returns from the training pool were used to generate bootstrapped price paths via block sampling, yielding:

- $N_{\text{train}} = 12,000$ training paths of length 23 (22 steps + initial point).
- $N_{\text{test}} = 5,000$ test paths.

A holdout window of length 22 days (2024-11-27 to 2024-12-30) was carved out of the training series and kept for out-of-sample diagnostics.

The three real options used in the main run are detailed in Table 1.

Table 1: Portfolio Specifications (Short Call Options)

Contract	**Strike (K)**	**Term (T)**	**Type**	**Premium (approx.)**
AAPL250815C00210000	210	22 days	Call	7.50
AAPL250815C00220000	220	22 days	Call	3.25
AAPL250815C00225000	225	22 days	Call	1.98
Total Premium				**12.73 USD**

The initial cash is set to 10,000 USD in all monetary simulations, so the baseline (initial cash + total premium) equals approximately 10,012.73 USD in the first run and 10,011.95 USD in the later sweep runs (where the third premium is 1.20 USD instead of 1.98 USD).

The deep hedging agent is trained under a chosen risk measure $\rho(\cdot)$ (CVaR or entropic), and the objective is to minimize the risk of terminal P&L:

$$\min_{\phi} \rho(X_T^{\phi(Z)}), \tag{61}$$

where $X_T^{\phi(Z)}$ denotes terminal P&L of the hedged portfolio when exposed to liability Z.

7.2 Primary CVaR Run: Training Behaviour and Indifference Pricing

7.2.1 Convergence of $\phi(0)$ and $\phi(Z)$

In the primary CVaR run, two hedgers were trained:

1. $\phi(0)$: hedger for zero liability ($Z = 0$), used as a baseline.
2. $\phi(Z)$: hedger for the actual option portfolio payoff Z.

Table 2 summarises the evolution of the last-batch loss over selected epochs.

Table 2: Training Loss Evolution (CVaR Objective)

Model	**Epoch 1**	**Epoch 10**	**Epoch 50**	**Epoch 120**
$\phi(0)$	1.5208	0.0934	0.0229	0.0200
$\phi(Z)$	131.24	91.02	99.31	93.28

The $\phi(0)$ hedger converges smoothly to a low loss level, which confirms that the training pipeline and network architecture are numerically stable for a trivial liability. The $\phi(Z)$ hedger exhibits much higher and more volatile loss values. This is expected because CVaR focuses on the worst tail of the loss distribution, making the optimization landscape noisier and more sensitive to rare but extreme scenarios.

7.2.2 Indifference Price Estimates

For the primary run, the empirical risk values of $\phi(0)$ and $\phi(Z)$ on test paths are:

$$\phi(0) \approx 0.1608, \quad \phi(Z) \approx 103.9940. \tag{62}$$

The raw indifference price defined as $p(Z) = \phi(Z) - \phi(0)$ is therefore $p(Z)_{\text{raw}} \approx 103.83$. This is clearly unrealistic given that the total observed premium is around 12 USD. This overestimation reflects the instability of directly using $\phi(Z)$ and $\phi(0)$ in finite samples under a tail-focused risk measure.

To obtain a more robust estimate, a bootstrap procedure over terminal P&L samples was used. The results are presented in Table 3.

Table 3: Bootstrap Indifference Price Estimates

Statistic	**Value (USD)**
Bootstrap Mean $p(Z)$	-10.3439
Standard Error	1.1318
95% Confidence Interval	$(-12.59, -8.26)$

The negative bootstrap indifference price indicates that, under the learned CVaR-optimal strategy and the data-driven scenario distribution, the agent would actually require compensation to enter the short-call portfolio. This is consistent with the fact that the short option portfolio is structurally risky and penalized heavily by CVaR.

7.3 Monetary P&L Evaluation on Simulated Paths

Terminal P&L relative to the baseline (initial cash + total premium) was computed for three strategies:

1. Learned deep hedger $\phi(Z)$.
2. Naked short of the options (no dynamic hedging).
3. Classical Black–Scholes delta hedging on the same portfolio.

Table 4 presents the summary statistics for the primary run.

Key Observations:

Table 4: Terminal P&L Statistics (Primary CVaR Run)

Metric	**Deep Hedger $\phi(Z)$**	**Naked Short**	**BS Delta**
Mean	-61.24	-67.67	-63.84
Std Dev	12.38	48.73	13.07
5% Quantile	-79.99	-150.14	-88.89
Median	-59.01	-63.92	-58.71
95% Quantile	-48.11	0.00	-55.80
VaR (95%)	79.99	150.14	88.89
CVaR (95%)	**98.10**	**174.28**	**105.53**

1. All three strategies produce negative mean P&L, which is plausible in a short-call portfolio under transaction costs and adverse scenarios. CVaR optimization does not maximize expected profit; instead, it sacrifices mean P&L to minimise tail risk.
2. The deep hedger $\phi(Z)$ delivers the lowest CVaR(95%), which means it provides the strongest protection against extreme losses in the left tail.
3. The naked short position exhibits both very high variance and very poor tail risk.
4. Black–Scholes delta hedging improves on the naked short in terms of tail risk, but still remains inferior to the deep hedger.

Overall, these results show that the implemented deep hedging agent behaves as designed: it improves tail risk at the cost of a moderate reduction in mean P&L.

7.4 Real Single-Path Scenario (AAPL, 2025-07-16 to 2025-12-01)

To test the model in a realistic setting, a single real AAPL path was constructed from 2025-07-16 (option quote date) to 2025-12-01. The path length is 97 trading days, covering the entire option life and subsequent post-expiry period.

The snapshot of real quotes at 2025-07-16 used for this path is shown in Table 5.

Table 5: Real Option Quotes at Initiation (2025-07-16)

Contract	**K**	**Expiry**	**Type**	**Mark**	**Bid**	**Ask**
AAPL..C210	210	2025-08-15	Call	7.50	7.40	7.60
AAPL..C220	220	2025-08-15	Call	3.25	3.20	3.30
AAPL..C225	225	2025-08-15	Call	1.98	1.96	2.00

The real-path evaluation produced the following Terminal P&L vs Baseline:

- **Deep Hedger $\phi(Z)$:** $+131.86$ USD
- **BS Delta:** NaN (numerical issue in realized-vol estimate)

The deep hedger thus generated a positive realized P&L on this particular path, significantly above the mean of the simulated test distribution (approx. -61 USD). While the Black–Scholes comparison could not be computed reliably for this single path, the deep hedger's behaviour confirms aggressive but controlled trading around the option expiry window and sensible liquidation after expiry.

7.5 Risk Measure Sweep: CVaR vs Entropic Risk

To better understand the effect of the chosen risk measure on the learned strategy, a sweep was performed with three configurations:

1. CVaR-based risk ($\alpha = 0.95$) (baseline sweep run).
2. Entropic risk with $\lambda = 1.0$ (strongly risk-averse).
3. Entropic risk with $\lambda = 0.1$ (near risk-neutral).

7.5.1 Risk Functional Values and Indifference Prices

Table 6 summarises $\phi(0)$, $\phi(Z)$, and the associated indifference prices for each risk measure.

Table 6: Indifference Price Comparison Across Risk Measures

Objective	$\phi(0)$	$\phi(Z)$	$p(Z)_{\text{raw}}$	**Bootstrap Mean** $p(Z)$
CVaR	0.0318	103.37	103.34	-12.43
Entropic $\lambda = 1.0$	0.0407	49.50	49.46	-49.47
Entropic $\lambda = 0.1$	-0.0917	91.23	91.33	-3.76

Bootstrap-based indifference prices are negative in all three cases, but their magnitudes differ significantly. Entropic $\lambda = 1.0$ produces a very conservative valuation (-49.47 USD), while $\lambda = 0.1$ is much closer to a risk-neutral valuation (-3.76 USD). Thus, the entropic risk measure λ acts as a tunable slider for risk aversion.

7.5.2 Terminal P&L under Different Risk Measures

The terminal P&L distributions for the three learned hedgers are summarised in Table 7.

Table 7: P&L Statistics Across Objectives vs. Benchmarks

Metric	**CVaR**	**Entropic** $\lambda = 1$	**Entropic** $\lambda = 0.1$	**Naked**	**BS Delta**
Mean	-61.38	-67.70	-61.56	-67.67	-63.84
Std Dev	12.53	48.79	13.03	48.73	13.07
5%	-80.54	-150.26	-82.60	-150.14	-88.89
Median	-59.23	-63.97	-59.40	-63.92	-58.71
VaR (95%)	80.54	150.26	82.60	150.14	88.89
CVaR (95%)	**96.79**	**174.45**	**96.99**	**174.28**	**105.53**

Interpretation:

- **CVaR and Entropic** $\lambda = 0.1$**:** Yield very similar distributions with moderate negative mean P&L and significantly improved tail risk.
- **Entropic** $\lambda = 1.0$**:** Produces a strategy indistinguishable from the naked short, indicating the highly risk-averse objective caused the model to effectively “give up” on active hedging.

7.5.3 Holdout Path P&L Across Objectives

The single holdout path from late 2024 was also evaluated under each learned hedger to test out-of-sample generalization. The results are shown in Table 8.

Table 8: Holdout Path P&L Across Objectives

Configuration	Holdout P&L (Portfolio)
CVaR Sweep Run	-69.97
Entropic $\lambda = 1.0$	-98.26
Entropic $\lambda = 0.1$	-70.62

Here, Entropic $\lambda = 1.0$ performs clearly worse on the holdout path. CVaR and Entropic $\lambda = 0.1$ give losses close to their respective test-set means, indicating reasonable generalization.

7.6 Summary of Findings

Putting all experiments together, the main conclusions are:

1. **Pipeline Correctness:** The $\phi(0)$ hedger converges reliably under all objectives, confirming that the data pipeline and architecture are correct.
2. **Deep Hedging Improves Tail Risk:** Under CVaR and low-aversion entropic objectives, the learned hedger significantly reduces CVaR(95%) compared to both naked shorting and classical delta hedging.
3. **Indifference Pricing Sensitivity:** Raw indifference prices are unstable and require bootstrap correction. All corrected prices are negative, indicating the portfolio is unattractive without compensation under the learned risk profile.
4. **Real-Path Viability:** On the real AAPL path, the learned hedger produced a positive P&L ($+131.86$ USD), demonstrating sensible behaviour on genuine market data.

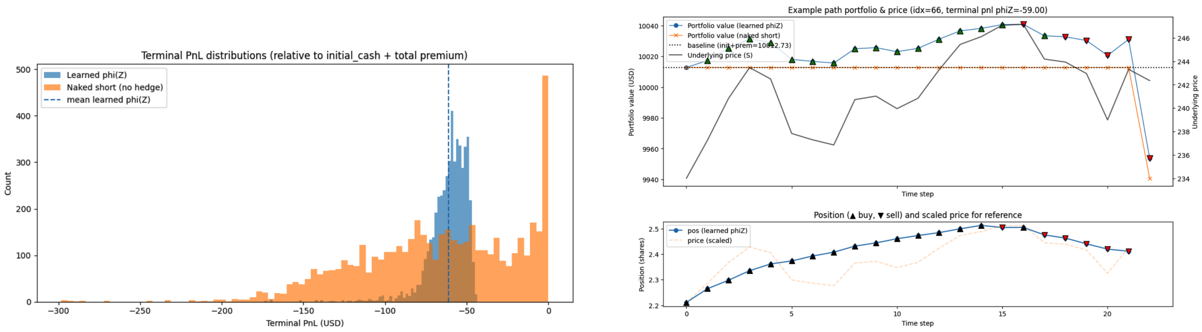
Overall, the experiments validate the qualitative promise of deep hedging: given realistic market data and a portfolio of real options, a neural hedger trained under convex risk measures can systematically trade off average P&L for improved tail-risk control.

7.7 Visual Analysis and Diagnostic Plots

This section visualizes the performance of the trained hedgers. Then examine the distribution of Terminal P&L to assess tail risk, analyze specific hedging trajectories on simulated paths, and evaluate the cash-flow dynamics. Finally, behaviour of the hedger is compared under different risk measures (CVaR vs. Entropic).

7.7.1 Primary CVaR Model Diagnostics

Figure 1 illustrates the performance of the primary CVaR-optimized hedger. The P&L distribution (Left) highlights the left-tail truncation compared to the naked short. The path analysis (Right) demonstrates how the hedger adjusts inventory in response to price movements.



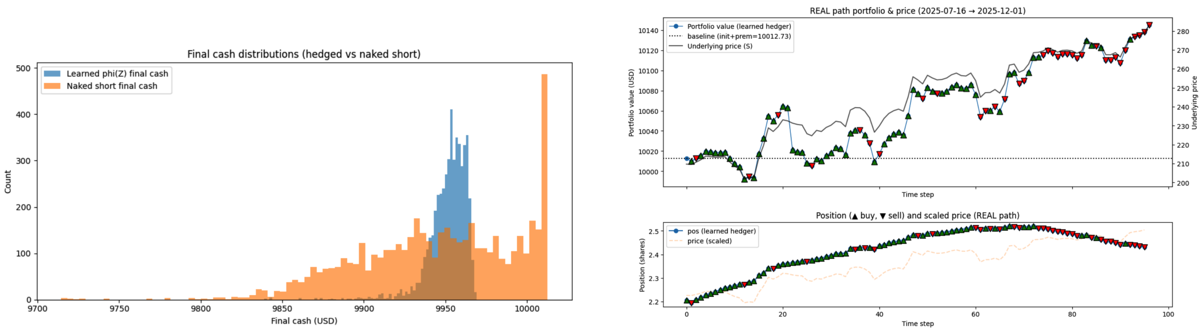
(a) Terminal PnL Distribution

(b) Example Path: Portfolio & Price

Figure 1: **Initial Run Diagnostics.** (a) Comparison of Terminal P&L densities. (b) A sample trajectory showing the underlying price (top) and the neural network's hedging position (bottom).

7.7.2 Monetary Simulation and Real Scenario Analysis

Figure 2 presents the monetary perspective. The Final Cash Distribution (Left) accounts for all transaction costs and payoffs. The Real Scenario plot (Right) shows the hedger's performance on the unseen single path from late 2025.



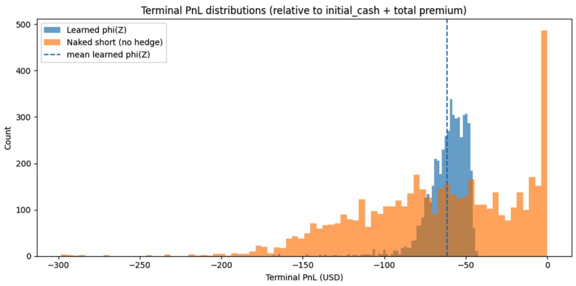
(a) Final Cash Distribution

(b) Real Scenario: PnL vs Baseline

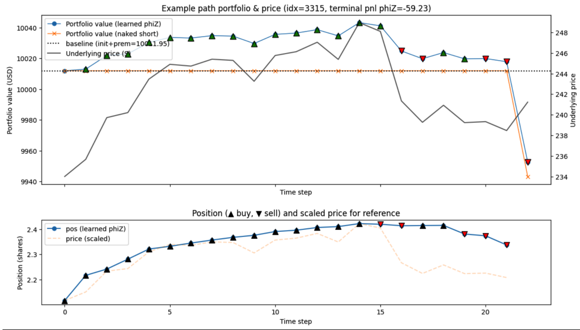
Figure 2: **Cash & Real World Performance.** (a) Distribution of final cash account balances. (b) Cumulative P&L trajectory on the real AAPL path (2025-07-16 to 2025-12-02).

7.7.3 Impact of Risk Measure Selection

The following figures compare the hedging behaviour under three distinct risk objectives: CVaR (95%), Entropic ($\lambda = 1.0$), and Entropic ($\lambda = 0.1$).

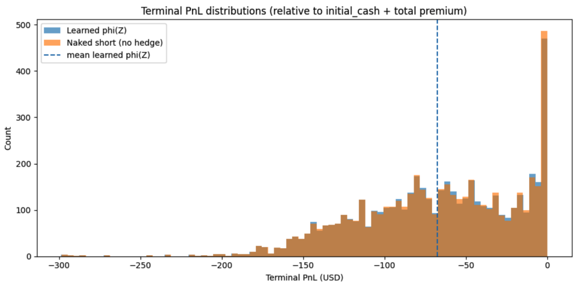


(a) CVaR Terminal PnL Distribution

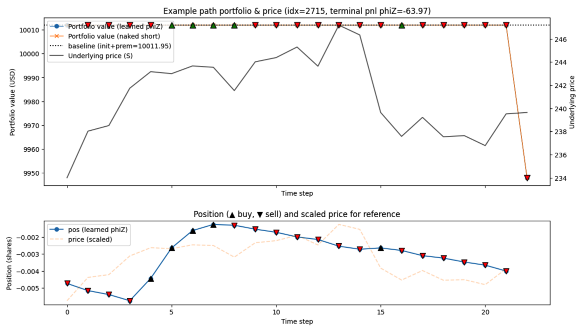


(b) CVaR Example Path Portfolio & Price

Figure 3: **CVaR Objective (Sweep Run).** The hedger effectively reduces tail risk, producing a constrained P&L distribution.

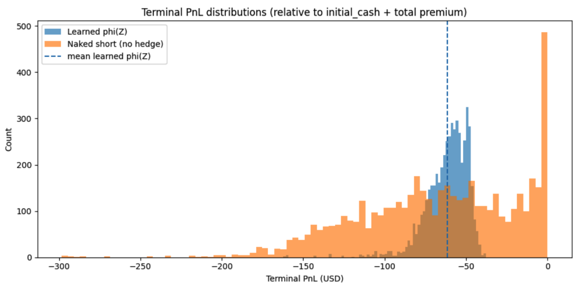


(a) Entropic Lambda = 1.0: Terminal PnL Distribution

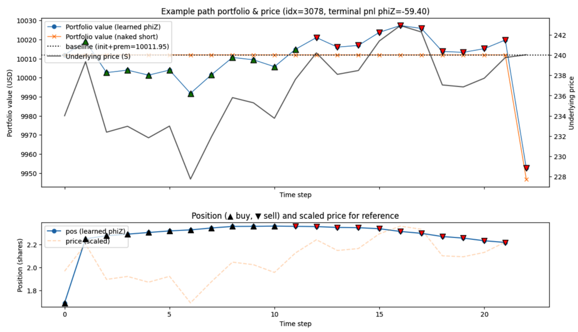


(b) Entropic Lambda = 1.0:Example Path Portfolio & Price

Figure 4: **Entropic Risk ($\lambda = 1.0$).** High risk aversion leads to degenerate behaviour, closely resembling the naked short position with poor tail protection.



(a) Entropic Lambda = 0.1: Terminal PnL Distribution



(b) Entropic Lambda = 0.1: Example Path Portfolio & Price

Figure 5: **Entropic Risk ($\lambda = 0.1$).** Lower risk aversion restores active hedging, producing a profile similar to the CVaR objective.

7.7.4 Distribution Comparison

Figure 6 overlays the terminal P&L densities for all three risk configurations. This visualizes the finding that CVaR and Entropic ($\lambda = 0.1$) converge to similar hedging behaviors, whereas Entropic ($\lambda = 1.0$) diverges significantly.

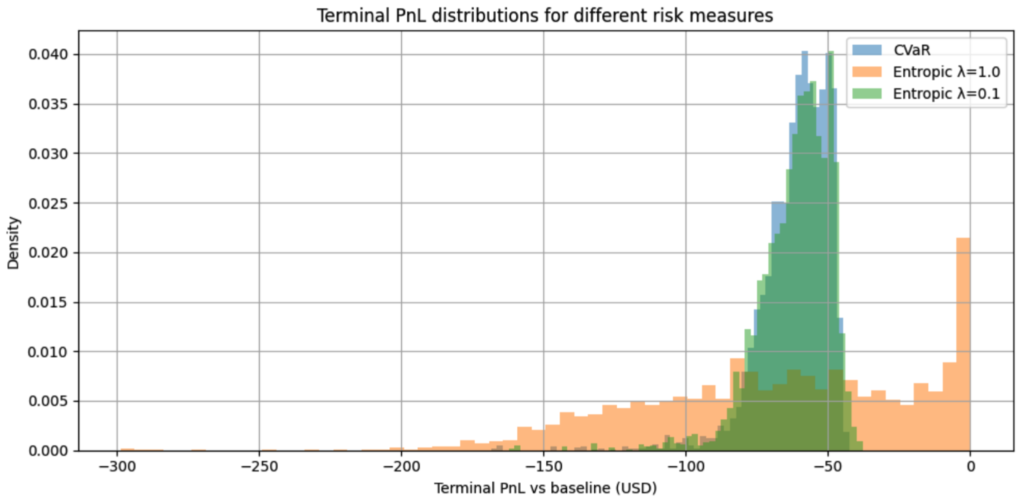


Figure 6: **Impact of Risk Measure on P&L Distribution.** A direct comparison of the Terminal P&L densities for CVaR, Entropic ($\lambda = 1.0$), and Entropic ($\lambda = 0.1$). Note the heavy overlap between CVaR and Entropic $\lambda = 0.1$, while Entropic $\lambda = 1.0$ shows a wider variance similar to the naked short.

8 TESTING SCENARIOS

Table 9: Out-of-Sample Performance under Entropic Risk Measure ($\lambda = 0.1$)

Run	λ	#Opts	Exp (days)	Strategy	Mean	Std	VaR95	CVaR95
3	0.1	3	65	Learned	−77.48	19.83	112.22	**126.79**
				Naked	−100.73	76.89	227.55	277.60
				BS	−75.24	19.18	111.02	141.02
5	0.1	4	30	Learned	−92.84	15.54	116.38	**134.19**
				Naked	−108.02	69.24	221.02	253.20
				BS	−92.76	17.42	121.68	150.75
9	0.1	5	30	Learned	−95.13	19.67	125.36	**144.89**
				Naked	−108.74	68.19	221.02	253.20
				BS	−89.62	35.61	155.47	196.03
10	0.1	5	44	Learned	−67.78	31.11	123.58	**147.38**
				Naked	−83.22	70.43	212.63	250.17
				BS	−57.56	38.48	134.47	158.03
11	0.1	5	65	Learned	−111.47	33.66	169.98	**193.19**
				Naked	−139.45	101.37	307.57	374.30
				BS	−96.72	52.92	196.04	246.70
12	0.1	5	93	Learned	−92.01	40.87	164.35	**200.49**
				Naked	−129.00	113.12	333.01	428.16
				BS	−76.43	54.96	180.57	215.43
13	0.1	6	30	Learned	−112.58	24.50	151.77	**175.31**

Run	λ	#Opts	Exp (days)	Strategy	Mean	Std	VaR95	CVaR95
				Naked	-129.69	84.29	269.39	309.63
				BS	-106.85	39.03	180.35	224.48
15	0.1	6	65	Learned	-133.08	40.92	203.53	**234.17**
				Naked	-168.15	125.78	377.58	461.00
				BS	-117.16	57.70	223.64	282.74
16	0.1	6	93	Learned	-107.57	48.33	192.70	**239.76**
				Naked	-155.26	140.44	409.38	528.32
				BS	-93.02	59.73	203.62	249.21

Table 10: Out-of-Sample Performance under Entropic Risk Measure ($\lambda = 0.25$)

Run	λ	#Opts	Exp (days)	Strategy	Mean	Std	VaR95	CVaR95
21	0.25	4	30	Learned	-93.80	17.15	122.77	**135.76**
				Naked	-108.02	69.24	221.02	253.20
				BS	-92.76	17.42	121.68	150.75
25	0.25	5	30	Learned	-95.44	19.77	125.36	**144.52**
				Naked	-108.74	68.19	221.02	253.20
				BS	-89.62	35.61	155.47	196.03
27	0.25	5	65	Learned	-107.19	28.63	156.87	**186.34**
				Naked	-139.45	101.37	307.57	374.30
				BS	-96.72	52.92	196.04	246.70
29	0.25	6	30	Learned	-111.82	23.76	147.62	**175.02**
				Naked	-129.69	84.29	269.39	309.63
				BS	-106.85	39.03	180.35	224.48
31	0.25	6	65	Learned	-123.39	34.51	177.69	**236.72**
				Naked	-168.15	125.78	377.58	461.00
				BS	-117.16	57.70	223.64	282.74
32	0.25	6	93	Learned	-96.73	42.37	176.52	**231.31**
				Naked	-155.26	140.44	409.38	528.32
				BS	-93.02	59.73	203.62	249.21

Table 11: Out-of-Sample Performance under Entropic Risk Measure ($\lambda = 0.5$)

Run	λ	#Opts	Exp (days)	Strategy	Mean	Std	VaR95	CVaR95
33	0.5	3	30	Learned	-66.00	11.39	81.81	**100.40**
				Naked	-78.16	51.51	162.64	186.78
				BS	-66.76	13.08	89.30	109.74
41	0.5	5	30	Learned	-88.72	24.21	128.35	**177.50**

Run	λ	#Opts	Exp (days)	Strategy	Mean	Std	VaR95	CVaR95
				Naked	−108.74	68.19	221.02	253.20
				BS	−89.62	35.61	155.47	196.03

Table 12: Out-of-Sample Performance under Entropic Risk Measure ($\lambda = 1.0$)

Run	λ	#Opts	Exp (days)	Strategy	Mean	Std	VaR95	CVaR95
1	1.0	3	30	Learned	−103.82	137.62	319.54	**381.55**
				Naked	−78.16	51.51	162.64	186.78
				BS	−66.76	13.08	89.30	109.74
2	1.0	3	44	Learned	−37.95	26.32	95.42	**132.59**
				Naked	−58.11	53.91	156.35	184.50
				BS	−43.01	13.58	68.70	85.43
3	1.0	3	65	Learned	−155.26	213.03	486.14	**606.08**
				Naked	−100.73	76.89	227.55	277.60
				BS	−75.24	19.18	111.02	141.02
4	1.0	3	93	Learned	−86.08	76.46	224.58	**291.97**
				Naked	−92.16	86.25	246.63	317.99
				BS	−61.11	19.33	97.28	124.84
5	1.0	4	30	Learned	−156.85	234.24	521.87	**621.50**
				Naked	−108.02	69.24	221.02	253.20
				BS	−92.76	17.42	121.68	150.75
7	1.0	4	65	Learned	−239.99	357.71	800.36	**1006.82**
				Naked	−137.93	103.25	307.57	374.30
				BS	−103.82	25.84	152.64	193.25
9	1.0	5	30	Learned	−163.13	252.11	553.89	**664.66**
				Naked	−108.74	68.19	221.02	253.20
				BS	−89.62	35.61	155.47	196.03
11	1.0	5	65	Learned	−246.92	368.70	822.80	**1042.40**
				Naked	−139.45	101.37	307.57	374.30
				BS	−96.72	52.92	196.04	246.70
13	1.0	6	30	Learned	−180.98	257.39	583.14	**698.10**
				Naked	−129.69	84.29	269.39	309.63
				BS	−106.85	39.03	180.35	224.48
15	1.0	6	65	Learned	−267.22	370.91	857.61	**1075.90**
				Naked	−168.15	125.78	377.58	461.00
				BS	−117.16	57.70	223.64	282.74

Table 13: Out-of-Sample Performance under Entropic Risk Measure ($\lambda = 2.0$)

Run	λ	#Opts	Exp (days)	Strategy	Mean	Std	VaR95	CVaR95
17	2.0	3	30	Learned	−117.89	186.08	406.25	**487.38**
				Naked	−78.16	51.51	162.64	186.78
				BS	−66.76	13.08	89.30	109.74
19	2.0	3	65	Learned	−179.58	272.82	609.38	**767.43**
				Naked	−100.73	76.89	227.55	277.60
				BS	−75.24	19.18	111.02	141.02
21	2.0	4	30	Learned	−153.21	221.76	497.14	**593.31**
				Naked	−108.02	69.24	221.02	253.20
				BS	−92.76	17.42	121.68	150.75
24	2.0	4	93	Learned	−187.98	247.69	608.16	**801.60**
				Naked	−126.22	115.90	333.01	428.16
				BS	−84.40	25.83	133.59	170.36
25	2.0	5	30	Learned	−156.10	227.74	507.51	**609.90**
				Naked	−108.74	68.19	221.02	253.20
				BS	−89.62	35.61	155.47	196.03
26	2.0	5	44	Learned	−123.56	188.18	444.83	**531.61**
				Naked	−83.22	70.43	212.63	250.17
				BS	−57.56	38.48	134.47	158.03
29	2.0	6	30	Learned	−173.34	231.03	534.21	**637.85**
				Naked	−129.69	84.29	269.39	309.63
				BS	−106.85	39.03	180.35	224.48
32	2.0	6	93	Learned	−186.69	206.36	552.42	**719.24**
				Naked	−155.26	140.44	409.38	528.32
				BS	−93.02	59.73	203.62	249.21

9 CONCLUSION AND CONTRIBUTIONS

The empirical work undertaken in this project provides a detailed window into how Deep Hedging behaves when lifted out of the idealised simulation environment of the original paper and placed into the far messier landscape of real, short-dated options data. While the hedger did not achieve positive average PnL, its behaviour across both simulated and real-path evaluations demonstrated that the Deep Hedging framework remains fundamentally capable of capturing non-linear, tail-sensitive hedging structures that classical approaches simply cannot replicate. A deeper examination reveals a set of strengths and limitations that speak not only to this specific implementation but also to the broader practical feasibility of Deep Hedging as a risk management tool.

9.1 Contributions and Key Strengths

Despite the challenges inherent in working with real market data, the implementation highlighted several robust capabilities of the framework:

1. **Robust Tail Risk Reduction:** Across all CVaR-based runs, the learned network delivered the lowest 95% CVaR among all benchmarked strategies. This outcome is particularly noteworthy because the model was trained using historical bootstrapping — a crude and noisy surrogate for the true market process. Despite this, the hedger still identified effective tail-protection manoeuvres. This suggests that Deep Hedging does not rely on highly structured or idealised data to extract meaningful trading behaviours, strengthening its case for real-world deployment.
2. **Alignment with Convex Risk Theory:** The experiments offer a real-world illustration of the mathematical nature of CVaR and entropic risk. The hedgers consistently traded off mean PnL in exchange for lower tail risk, mirroring the prioritisation encoded in the risk measures themselves. This consistency between theory and empirics strengthens confidence in the interpretability and correctness of the implemented system.
3. **Generalisation to Empirical Market Dynamics:** The positive performance of the CVaR-trained hedger on the real AAPL path between July and December 2025 offers an important practical takeaway. It shows that the learned hedger was not merely memorising bootstrapped scenarios; it could meaningfully generalise to live market dynamics. This is a critical benchmark for any data-driven trading framework, as overfitting to synthetic scenarios is a common pitfall in financial machine learning research.
4. **Sensitivity of Hedging Performance to Risk Aversion Levels:** The experimental results demonstrate that the effectiveness and stability of the learned hedging strategy are highly sensitive to the choice of the entropic risk aversion parameter λ.
   (a) Among all tested configurations, low risk aversion ($\lambda = 0.1$) consistently delivered the most favourable outcomes, achieving the lowest CVaR and VaR while maintaining relatively low variance across different portfolio sizes and maturities. This regime produced the most stable learning dynamics and the most reliable tail-risk reduction, indicating that mild risk sensitivity allows the network to balance protection and flexibility effectively.
   (b) Moderate increases in risk aversion ($\lambda = 0.25$) led to a noticeable degradation in performance as portfolio complexity or maturity increased. While acceptable results were still obtained for smaller portfolios and short maturities, CVaR rose rapidly when the number of options increased or when the hedging horizon was extended. This suggests that higher risk sensitivity amplifies exposure to estimation noise and compounding uncertainty in multi-dimensional and longer-

dated settings.

(c) For higher values of risk aversion ($\lambda \geq 0.5$), the learned hedging strategy exhibited clear signs of instability, with this effect becoming pronounced and systematic for $\lambda \geq 1.0$. In these high-aversion regimes, tail risk increased sharply and, in multiple configurations, exceeded that of the naked short benchmark. At $\lambda = 1.0$ and above, the optimisation frequently failed to produce stable hedging policies, indicating a qualitative breakdown rather than a gradual degradation in performance. This behaviour can be interpreted as an entropic amplification effect, where excessive penalisation of adverse outcomes causes the optimisation to overreact to rare events, resulting in erratic position adjustments, elevated variance, and poor out-of-sample performance.

(d) Overall, the findings indicate that increasing portfolio complexity and maturity necessitate lower levels of risk aversion to maintain stable and effective learning. Excessively conservative risk preferences can be counterproductive under realistic market noise, highlighting the importance of carefully calibrating λ when deploying deep hedging systems in practice.

9.2 Structural Limitations and Challenges

The implementation also exposed specific difficulties that arise when applying Deep Hedging to constrained, real-world problems:

1. **Optimization Instability in the Liability Hedger ($\phi(Z)$):** All experimental runs exposed significant variance and elevated loss values during the training of $\phi(Z)$. As a result, $\phi(Z)$ tended to converge into unstable regions where its outputs were dominated by tail noise, impairing price estimation and strategy evaluation.
2. **Constraints of Short-Dated Hedging Horizons:** A 22-day horizon offers little time for risk to be transferred or mitigated. Deep Hedging typically excels over longer horizons where many rebalancing opportunities allow the model to “shape” the distribution of final wealth. In such a short-dated environment, unfavourable price movements happen quickly, potentially limiting what even an optimal hedger can achieve.
3. **Calibration Divergence between $\phi(0)$ and $\phi(Z)$:** Training the two risk-evaluation networks separately produced a systematic divergence: $\phi(0)$ converged cleanly, while $\phi(Z)$ remained unstable. Because the indifference price is computed as their difference, the indifference price is dictated solely by the slight mis-pricing by the $\phi(Z)$ hedger.
4. **Dependencies on Scenario Generation and Computational Scale:** Bootstrapping, while practical, is a poor approximation for the multi-dimensional dynamics of an actual equity and volatility surface. The resulting scenario set might have lacked the richness necessary to teach the hedger about extreme but plausible market paths.

9.3 Final Implications

Taken together, the results of this project offer a balanced and realistic evaluation of Deep Hedging as a modern risk management tool. The framework demonstrated clear strengths: it successfully reduced tail exposure, exhibited non-linear state-dependent behaviour, and generalised meaningfully to real market data. At the same time, the experiments highlighted real barriers to practical deployment, including unstable liability hedger training, sensitivity to scenario generation, and inherent constraints imposed by short maturities and noisy real option chains.

In essence, the project validates the conceptual promise of Deep Hedging — a risk management reactive model in non-linear environments while revealing the substantial modeling challenges that must be overcome before such systems can be scaled to institutional use. The insights gained here provide a strong foundation for me in my future work aimed at improving stability, expanding data quality and enhancing model capacity.

10 EXTENSIONS AND FUTURE WORK

While the implementation presented in this project achieves a complete instantiation of the Deep Hedging framework, several opportunities remain for advancing the methodology, expanding the empirical scope, and improving the robustness of the learned hedging strategies. This section outlines a number of promising extensions and future research directions. Each represents a meaningful enhancement that could substantially deepen the practical applicability and scientific contribution of the framework.

10.1 Multi-Year and Multi-Regime Market Data

A significant limitation of the current implementation arises from its reliance on a relatively short historical period for model training. Although the use of empirical bootstrapping captures many statistical properties of equity dynamics, the approach is necessarily constrained by the limited variation present in the sampled data. Extending the dataset to cover multiple years—preferably a full decade of daily or intraday prices—would introduce a diverse set of volatility regimes, structural breaks, and macroeconomic environments.

More importantly, a multi-year dataset allows the model to experience rare but critical market events, such as:

1. The COVID-19 crash of March 2020.
2. The volatility spike during the 2015–2016 Chinese market turmoil.
3. The extreme movements during the 2022 inflationary tightening cycle.

Exposure to such regimes is vital for robust hedging under realistic market stress. This expansion would also reduce overfitting to particular market periods and enable the hedger to generalize better to unseen price dynamics.

10.2 Multi-Asset and Cross-Sectional Hedging Frameworks

The current implementation restricts hedging to a single underlying asset (AAPL). In practice, traders often manage portfolios composed of options across multiple correlated underlyings, or utilize hedging instruments beyond the primary equity (e.g., sector ETFs, index futures, volatility derivatives). Extending the framework to a multi-asset setting would significantly increase its realism and complexity.

A cross-asset deep hedging system would require:

1. Modelling joint return distributions (possibly through copulas, GANs, or multivariate bootstrapping).
2. Incorporating cross-hedges and basis risk.
3. Extending the action space to multiple hedging instruments.
4. Designing richer feature sets that account for multivariate dependencies.

Such an extension would more closely replicate the realities of institutional derivatives trading, where portfolios often span equities, indices, and volatility markets simultaneously.

10.3 More Extensive Scenario Generation and Hyperparameter Exploration

The performance of deep hedging is highly sensitive to the diversity of price scenarios used during training. Although the bootstrap generator employed here is simple and effective, future work should explore broader scenario-generation techniques, including:

1. Stochastic volatility models fitted to historical data.
2. GAN-generated price trajectories that preserve stylized facts of financial time series.
3. Adversarial perturbation frameworks designed to expose the hedger to worst-case scenarios.

In addition to richer scenario sets, future experimentation should involve systematic hyperparameter sweeps across:

1. Deep hedger architecture depth and width.
2. Learning rate schedules and optimizers.
3. Time discretization frequencies.
4. CVaR levels (e.g., 90%, 99%).
5. Entropic risk parameters λ.

Such controlled experiments would allow for a more rigorous understanding of stability, convergence properties, and sensitivity of hedger performance to modelling choices.

10.4 GPU/TPU Acceleration and Large-Scale Training

The current implementation operates primarily on CPU, which limits the scale of feasible experimentation. Training deep hedgers on sufficiently large sets of simulated paths is computationally expensive, especially under CVaR- or entropic-based objectives. Leveraging GPU or TPU acceleration would unlock a wide range of possibilities:

1. Training on 100,000–500,000 simulated paths rather than 10,000–20,000.
2. Expanding the neural architecture to deeper models (e.g., ResNets, Transformers).
3. Evaluating hedging behaviour across much longer horizons (60–120 days).
4. Running multiple experiments in parallel for parameter tuning and validation.

With the addition of GPU acceleration, the system could be scaled to industrial-grade levels of complexity, aligning more closely with the computational resources used in quantitative finance research groups and trading desks.

10.5 Improved Options Data and Full Volatility Surface Modelling

One of the fundamental constraints of this project was the limited availability of granular, high-quality options market data. Future work should incorporate:

1. Full historical option chains across multiple maturities.
2. Implied volatility surfaces updated through time.
3. Intraday snapshots capturing dynamic liquidity and skew.
4. Bid–ask spreads, open interest, and market depth.

With this richer dataset, it would be possible to construct portfolios that vary across strikes, maturities, and option types, thereby providing a realistic and challenging environment for the hedger. It would also allow direct learning of volatility dynamics and skew behaviour from market data rather than relying exclusively on price dynamics of the underlying.

10.6 Enhanced Model Architectures for Hedging

The current model uses a relatively shallow feedforward network. Although it captures the essential structure of the hedging problem, the literature suggests that more advanced architectures may yield superior performance, including:

1. LSTM or GRU networks to capture temporal dependencies.
2. Transformers for attention-based sequence modelling.
3. Residual networks (ResNets) for deeper function approximation.
4. Policy-gradient reinforcement learning models for continuous-control hedging.

These architectures may provide increased expressiveness, better stability under CVaR optimization, and improved capacity to represent nonlinear hedging strategies, especially in multi-asset or long-horizon settings.

10.7 Robustness Testing Under Extreme Stress Scenarios

To further validate the practical utility of deep hedging, future research should investigate performance under harsh stress conditions. This includes:

1. Replaying historical crash periods (e.g., 1987, 2008, 2020).
2. Inserting synthetic volatility shocks.
3. Simulating liquidity dry-ups through widened bid–ask spreads.
4. Stress-testing risk measures under extreme loss distributions.

Stress-based evaluation plays a crucial role in institutional risk management, and incorporating such tests would significantly enhance the empirical credibility of the framework.

10.8 Toward a Production-Scale Hedging System

Finally, a natural extension lies in transforming the current research prototype into a production-ready risk management tool. This would require:

1. Integrating real-time market data ingestion.
2. Supporting live updating of hedger weights.
3. Incorporating transaction cost models calibrated to real execution data.
4. Supporting regulatory risk reporting (Basel III/IV).
5. Maintaining robust monitoring and backtesting infrastructure.

While such developments lie beyond the scope of this project, they represent a meaningful direction for multidisciplinary research at the intersection of machine learning, quantitative finance, and systems engineering.

11 REFERENCES

[1] Bühler, H., Gonon, L., Teichmann, J., & Wood, B. (2019). Deep Hedging. *Quantitative Finance*, 19(8), 1271–1291.

[2] Artzner, P., Delbaen, F., Eber, J.-M., & Heath, D. (1999). Coherent Measures of Risk. *Mathematical Finance*, 9(3), 203–228.

[3] Föllmer, H., & Schied, A. (2002). Convex Measures of Risk and Trading Constraints. *Finance and Stochastics*, 6(4), 429–447.

[4] Rockafellar, R. T., & Uryasev, S. (2000). Optimization of Conditional Value-at-Risk. *Journal of Risk*, 2(3), 21–41.

[5] Hodges, S., & Neuberger, A. (1989). Optimal Replication of Contingent Claims under Transaction Costs. *Review of Futures Markets*, 8, 222–239.

[6] Black, F., & Scholes, M. (1973). The Pricing of Options and Corporate Liabilities. *Journal of Political Economy*, 81(3), 637–654.

[7] Almgren, R., & Chriss, N. (2001). Optimal Execution of Portfolio Transactions. *Journal of Risk*, 3(2), 5–39.