Analysis of the Quantum Advantages for Deep Hedging

Soham Deshpande
Srinandan Dasmahapatra

October 2024

Abstract

Parameterised Quantum Circuits (PQCs) have opened many doors, one such being the use in financial markets. In this paper, I look at the problem of optimally hedging a portfolio of derivatives through the use of quantum computing. Given a Quantum Circuit Born Machine (QCBM), we are able to generate synthetic data that mimics the statistical distribution of the original dataset. We are generating synthetic market data, here the dynamics of a given vanilla option is learnt. The market generator is then used to simulate the option to maturity with the intent of learning an optimal hedging strategy π^*, a showcase of a data-driven approach to risk hedging. Different generator architectures will be compared to maximise the quality of the synthetic data. The findings of this research will contribute to the growing literature on risk management in quantitative finance, with applications of the market generator extending beyond deep hedging.

Contents

1 Problem Statement

The problem of hedging a portfolio of derivatives is an important part of risk management used widely in financial institutions. We can picture a perfect, frictionless market where transaction costs are negligible and every asset in the space has a price. Here we can price and hedge perfectly. Unfortunately in practice, we experience incomplete markets due to frictions. In recent years markets have experienced periods of heightened volatility, much that disobey traditional frameworks. This generates the need for complex, realistic market models that can account for these.

Traditional methods of pricing options has shown to be ineffective for equity markets, new information resulting in rapid changes. Much of the available literature models the market as a smooth, continuous stochastic process within a Gaussian space. Such models are sufficient for common market activity but fail when presented with discontinuous moves in price. In the context of Euro Stoxx 50, these can be reactions to geopolitical events or natural disasters; traditional models being incapable of capturing the effects. The introduction of Jump-diffusion models aimed to solve this issue though faces similar issues. In reaction we have recently observed non-parametric models that harness neural networks and machine learning which aim to demonstrate high accuracy on out-of-sample forecasts.

In the deep hedging framework, we are able to construct a dynamic optimal hedging policy by following realistic market simulations to maturity. This is a policy that involves a mapping from market states to actions. Instead of solving analytical equations under unrealistic market assumptions, such as Greek hedging, we can instead model the trading decisions made as an optimisation problem, particularly suitable for neural networks with a rich feature set. This will be done by focusing on the accuracy of the market simulation itself; I will aim to utilise quantum computing techniques to enhance market simulations so that a more optimal hedging strategy can be learnt than prior techniques. In this, an exploration into quantum circuit architectures will be conducted, as well as exploiting quantum machine learning for the simulation and policy search.

2 Related Literature

To place this research within the context of existing literature, we can split the project into 2 components: the market generator, and quantum parameterised circuits.
The work around deep hedging has evolved with time, moving away from greek-based and statistical hedging towards a more sturdy framework using machine learning. Here a lot of work is being done, with many papers emphasising on using neural networks for optimising delta and gamma exposure. [**Enhancing BS delta**, **Deep Gamma Hedging**]. Beuhler introduced the initial approach, modeling trading decisions as neural networks instead of relying on parameterised models. [**Beuhler Deep Hedging**] Many advancements were made after this with researchers focussing on developing realistic market simulators. Wissel proposed a market model for path generation of options but this still employed risk-neutral diffusion. Wiese then introduced a new dimension by using GANs by converting options into local volatility models with simpler no-arbitrage constraints. This focussed on the local stochastic nature of options. [**Learning to simulate equity markets**] Some approaches suggest using actor-critic reinforcement learning algorithms to solve for an optimal value function, a move towards searching for a global maximum instead of thinking about local risk management.[**deep bellman**]
In recent times, we have also seen research done on using quantum computing to hedge portfolios, here the authors presented a quantum reinforcement learning method based on policy-search and distributional actor-critic algorithms. They proposed using a Quantum Neural Network(QNN) to approximate the value of a given value function by predicting the expected utility of returns. They constructed this using compound and orthogonal layers which were built using Hamming-weight unitaries. This helped overcome the barren plateau by ensuring the gradient variance does not vanish exponentially with qubit count. Another method was also proposed by modeling the entire distribution of returns rather than just the expectation. By exploiting the power of parameterised circuits, the categorical distribution was learnt enabling it to capture variability and tail risk. [**quantum deep hedging**]
There is an immense amount of research being done on exploiting the benefits of quantum computing, recent advancements being in quantum algorithms. These claim to provide exponential speed-up over classical methods, though in reality we see great complexity in state preparation, requiring $\Theta(2^n/n)$ circuit depth with n qubits or $\Theta(n)$ circuit depth with $\Theta(2^n)$ ancillary qubits. Here we see hybrid models such as Born machines and Quantum generative adversarial networks boasting high generalisation ability. [**do quantum circuits generalise**]
There has also been research in harnessing back action from quantum weak measurements to enhance the ability of quantum machine learning algorithms. In quantum reservoir computing, the ability to retain information from past inputs plays a key role in processing temporal series and producing future predictions.[10]
This research aims to combine the needs of financial firms in hedging portfolios using realistic market models by utilising QCBMs as a tool for simulating paths in combination with quantum neural networks for learning optimal policies. A comparison will be made against hedging under Black-Scholes.

3 Markets and Derivatives

The market, though inherently can be thought of as a completely random process, where bids and asks are fulfilled, can be modelled as a stochastic process. The aim of this chapter is to serve as a brief introduction and set up notation for later chapters.

3.1 Market

Consider a market with a finite time horizon T defined on the probability space $(\Omega, \mathcal{F}, P)$ along with a filtration $\mathbf{F} = \{\mathcal{F}|0 \leq t \leq T\}$ This can be thought of as an adapted (n+1) dimensional Itô process $X(t) = (X_0(t), X_1(t), ..., X_n(t))$ which has the form

$$dX_0(t) = \rho(t,\omega)X_0(t)dt; \quad X_0(0) = 1 \tag{3.1}$$

and

$$dX_i = \mu_i(t,\omega)dt + \sigma_i(t,\omega)dB(t); \quad X_i(0) = x_i \tag{3.2}$$

where $X_i(t)$ as the price of asset i at a given time t.
We can define a portfolio in the market as

$$\theta(t,\omega) = (\theta_0(t,\omega), \theta_1(t,\omega), ..., \theta_n(t,\omega)) \tag{3.3}$$

where the components $\theta_n(t,\omega)$ represents the number of units of a given asset held at time t.
Following from that, we can define the value of a given portfolio to be

$$V(t,\omega) = V^{\theta}(t,\omega) = \sum_{i=0}^{n} \theta_i(t)X_i(t) \tag{3.4}$$

Lastly, it is important to state that the portfolio is self-financing, any trading strategy α requires no extra cost beyond the initial capital

$$V(t) = V(0) + \int_0^t \theta(s) \cdot dX(t) \tag{3.5}$$

We can also make the following assumptions about the market:

- The market is liquid, allowing the trade to execute instantaneously
- There is no bid-ask spread, the price to buy and sell is the same
- Trading actions taken have no impact on the price of the asset traded

3.2 Derivatives

A derivative refers to any financial instrument whose value is derived from an underlying security, the most fundamental being futures and options.

3.2.1 Futures

A futures contract is a contract that gives the right and obligation to buy a given asset i a specified time T at price K. This can be thought of as the underlying asset for an options contract.

3.2.2 Options

The two types of options going to be explored are Puts and Calls; a Call option gives the owner the right but not the obligation to buy a given asset i at a specified price K at time T. Similar to the Call, a Put option gives the owner the right but not the obligation to sell a given asset i at a price K at time T. If the owner can exercise the option any time up to T, we call this an American option. For the purposes of this research, we will only be dealing with Vanilla European options.
It is important to define the payoffs for both options:

$$C_T = max(0, S_T - K) \tag{3.6}$$

$$P_T = max(0, K - S_T) \tag{3.7}$$

3.3 Euro Stoxx 50

In this research I will be focussing on a single asset, the Euro Stoxx 50 Index (SX5E) and relevant derivatives. This is a stock index of 50 stocks in the Eurozone. This index captures around 60% of the free-float market capitalisation of the Euro Stoxx Total Market Index which covers about 95% of the free-float market in the Eurozone. Rational behind choosing this index is the availability of data, options traded with it as the underlying and the liquidity of the index.
Derivatives that are held in the portfolio to be hedged will include those that have SX5E as the underlying, examples are weekly, monthly, and quarterly expiration options. These are European-style so can only be exercised upon maturity.

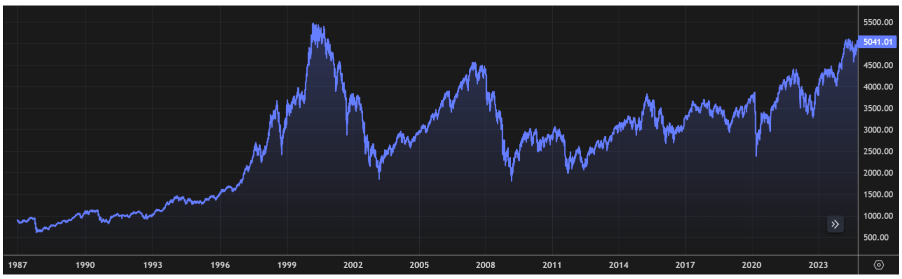


Figure 1: Euro Stoxx 50 price chart

4 Deep Hedging

Hedging can be thought of as a stochastic control problem where we have 3 actors, a market state, randomness from market returns and a hedging strategy. By making observations on the market state we are required to make adjustments to one's hedging policy. We can formally define hedging as :

$$\min_{\theta \in \Theta} \rho(H^\theta - P)$$

where Θ is the set of admissible hedging strategies, P is the uncertain payoff of derivative at final time, H^θ is the uncertain payoff of hedging portfolio if strategy θ is used at final time, and ρ is a risk measure.

It can be assumed that the hedging instrument used, SX5E in this case, can be traded with sufficient liquidity and that the trading of the asset does not influence the price. Given these conditions, we can formulate the problem of deep hedging to be a reinforcement learning one.

We can set up two sets of rewards, R, positive from the cash flows from the portfolio of hedging instruments and derivatives. The second is C, the negative reward associated with transaction costs. For simplicity, we can assume that the cost is proportional to the cost of the hedging instrument. By taking a set of actions at each time step, t, we can formulate a trading strategy which gives an expected return of $T_t^\pi = (R_{t+1}^\pi - C_{t+1}^\pi) + ... + (R_T^\pi - C_T^\pi)$. The goal of the reinforcement algorithm would be to maximise the expected return. However, to create a strategy that does not incur too much risk, it is important to include metrics such as Value at Risk (VaR) and Conditional Value at Risk(CVAR). These aim to minimise the tail-risk involved in a given trading strategy hence creating a more robust strategy and conforming to regulations such as Basel III. [**The New International Regulation of Market Risk**]

One of the main problems is the scarcity of data, calibration of such models requires huge amounts of realistic market data for the underlying. Hence this requires a market generator. Traditionally a GBM with Monte Carlo simulations would be used but this paper goes beyond, using quantum computing to combat the aforementioned issues.

The design of the neural network responsible would involve an input layer which takes the market state such as the underlying, and options features such as the strike price and maturity.

The loss function will have type: Loss $= \lambda_1 \cdot \mathcal{L}_R + \lambda_2 \cdot \mathcal{L}_{CVaR}$ where λ_i is used to define a weighting to the two losses, $\mathcal{L}_R$ is the replication loss from simulating the options to maturity given a generator. We can define it as $\mathcal{L}_R(X_T^\alpha - g(S_t))$ where X_T^α is the portfolio at maturity and $g(S_t)$ is the option pay-off.

VaR can be defined as :

$$\mathrm{VaR}_\alpha(X) = \inf\{x \in \mathbb{R} : P(X \leq x) \geq \alpha\} \tag{4.1}$$

CVaR likewise:

$$\mathrm{CVaR}_\alpha(X) = \frac{1}{1-\alpha} \int_{-\infty}^{\mathrm{VaR}_\alpha(X)} x \cdot p(x)\, dx \tag{4.2}$$

This takes a global approach towards finding an optimal policy instead of managing a local criterion as suggested by [**Fecamp**].

5 Parameterised Circuits

A basic understanding of quantum mechanics and computing is assumed.

5.1 Born Rule

An essential part of modern quantum computing involves the existence of the Born Rule. Born's measurement rule states that:

$$p(x) = |\langle x|\psi(\theta)|\rangle|^2 \tag{5.1}$$

The state $|\psi(\theta)\rangle$ is generated by evolving the vacuum state $|0\rangle$ according to a Hamiltonian H that is constructed from gates. Once combined, the gates form a parameterised quantum circuit which is parameterised by using the variables governing each gate, θ. By tuning the values of θ_i one can allow for an evolution to any state that will serve as a solution to a given problem.
By taking the distribution associated to the state, $|\psi(\theta)\rangle$ we can treat the PQC as a generative model, which upon measuring in a given basis, will generate samples of a target distribution χ. This model is parameterised by θ, which defines a quantum circuit $U(\theta)$ made up of a set of quantum gates such that:

$$|\psi(\theta)\rangle = U(\theta)|0\rangle^{\otimes n} \tag{5.2}$$

By measuring the circuit, we can obtain samples. Producing samples that emulate the target distribution involves minimising the parameters of the circuit $U(\theta)$, a process once convergence is reached, will generate accurate samples. [**bornmachine**]

5.2 State Preparation

State preparation is a core part of the hybrid quantum computing model. This concerns the most efficient methods to input classical data into a quantum system.
Without the use of ancillary qubits, we can expect an exponential circuit depth to prepare an arbitrary quantum state. Using them we can reduce the depth to be sub-exponential scaling, with recent advancements reaching $\Theta(n)$ given $O(n^2)$ ancillary qubits. [**stateprep1**] [**stateprep2**] We require state preparation to transfer the classical data, bits, onto the Hilbert space. This involves a function ϕ that maps the input vector to an output label. There are many encoding schemes, each of which aim to offer high information density and low error rates. The main methods of state preparation include: basis, amplitude, angle encoding, and QRAM.

5.2.1 Amplitude Encoding

Amplitude encoding consists of the following transformation:

$$S_X|0\rangle = \frac{1}{||x||}\sum_{i=1}^{2n} x_i|i\rangle \tag{5.3}$$

where each x_i is a feature of the data point x, and $|i\rangle$ is the basis of the n-qubit space. This does boast the advantage of being able to store 2^n features using only n qubits but does create a resultant circuit with depth $O(2^n)$.

5.2.2 Angle Encoding

The transformation used is:

$$S_x |0\rangle = \otimes_{i=1}^{n} cos(x_i) |0\rangle + sin(x_i) |1\rangle \tag{5.4}$$

It can be constructed using a single rotation with angle x_i which is normalised to be within $[-\pi, \pi]$. This allows us to store n features with n qubits.
A possible encoding method is dense angle encoding which allows us to encode $2n$ features in n qubits but hasn't been chosen in this project for simplicity; the same can be said for amplitude encoding.

5.3 Measurements

In quantum mechanics, we can define measurement to be any process that probes a given quantum system to obtain information, we may also refer to this process as a measurement in the computational basis when focussing on quantum information science. Let's consider a quantum state $|\psi\rangle = \alpha_0|0\rangle + \alpha_1|1\rangle$ which gives us $|0\rangle$ with probability $|\alpha_0|^2$ and $|1\rangle$ with probability $|\alpha_1|^2$. If measured in the standard basis we would expect the outcome to be $|k\rangle$ (for k = 0,1)with a given probability. This outcome would result in the output state of the measurement gate to also be $|k\rangle$, resulting in the original state $|\psi\rangle$ to be irreversibly lost. We can refer to the process the state undergoes as a collapse of state.

5.4 Quantum Circuit Born Machine

Given a dataset $D = \{x_1, x_2..x_n\}$ consisting of n samples and obeys a given distribution χ_d, we would like the QCBM to learn the distribution and generate synthetic data points that are of the distribution χ_s such that χ_s approximates χ_d. The Quantum circuit Born machine is a subclass of parameterised quantum circuits, here the quantum circuit contains parameters which are updated during a training process. The QCBM takes the product state $|0\rangle$ as an input and through an evolution, transforms into a final state $|\phi_0\rangle$ by a sequence of unitary gates. This can then be measured to obtain a sample of bits $x \sim p_\theta(x_s) = |\langle x| |\phi_\theta\rangle|^2$. By training the model we are aiming to let p_θ approach χ_d.
The ansatz for the quantum circuit used by the Born machine consists of 7 layers of 1-qubit gates with entangling layers in between them. These are entangled using the CNOT gates as shown in figure x. The number of wires(qubits) needed depends on the precision required for the generated data. The estimated precision is 12-bit, so the data being able to take 2^{12} different values in the range of $(v_{min} - \epsilon, v_{max} + \epsilon)$, where $\epsilon > 0$ allows data to be generated that lie outside the range (v_{min}, v_{max}) of the original data.
The QCBM takes a $n \times m$ matrix of parameters in the range $(-\pi, \pi)$ as input, in the form of a dictionary. Each angle takes one of 2^k discrete values, where k is a model parameter. The resulting space therefore spans to: $(2^m)^{n \cdot m}$

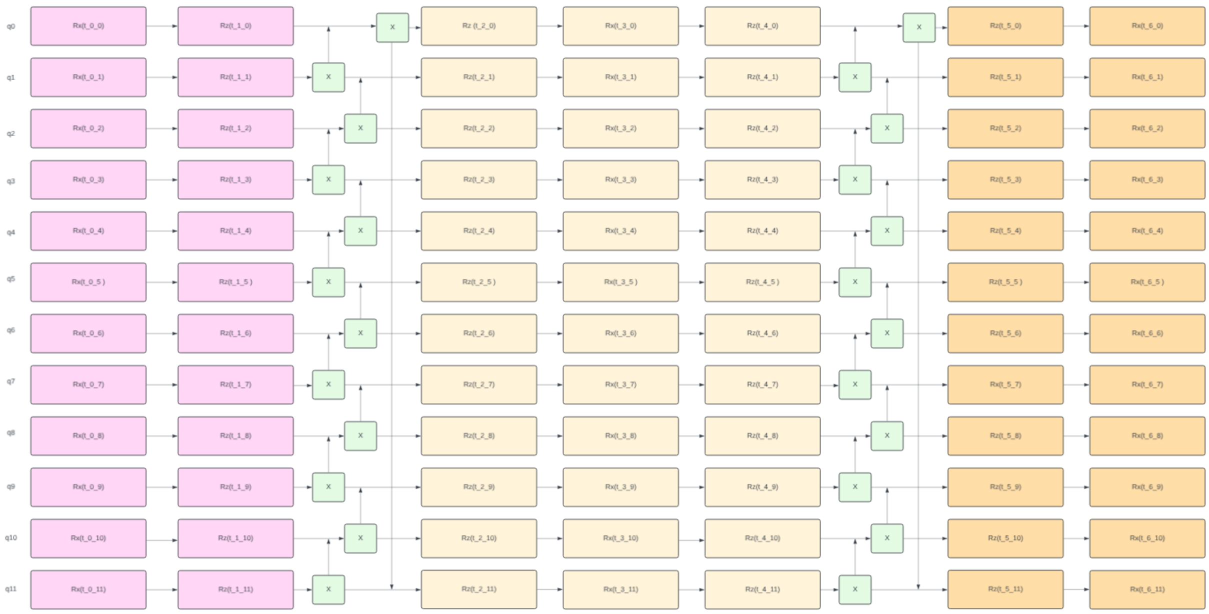


Figure 2: QCBM architecture

5.4.1 Testing

Many methods were explored to generate time-series data from the QCBM as well as testing parameters of the model. The first idea explored involved learning the distribution of the delta value of an American option.

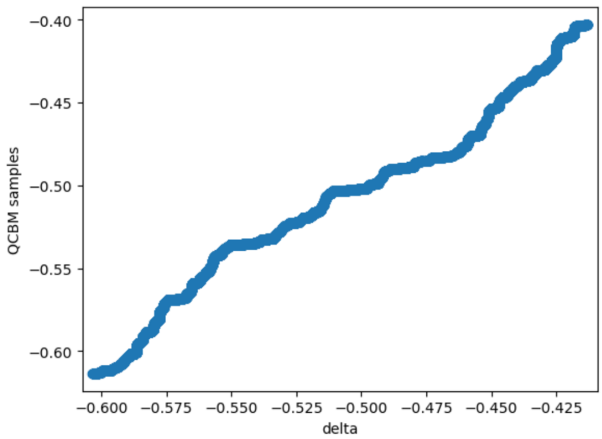


Figure 3: Quantile plot for American option delta

Given the delta of an option through a day, the QCBM demonstrated high generalisation power as shown by the diagonal nature of the plot. A genetic algorithm was employed to learn the set of parameters θ.
The circuit also provided promising returns when learning the vega of the same option. It was also important to verify the generalisation by learning the strike price, which was a constant for the option. We can observe the two histograms of the samples generated from measurement.

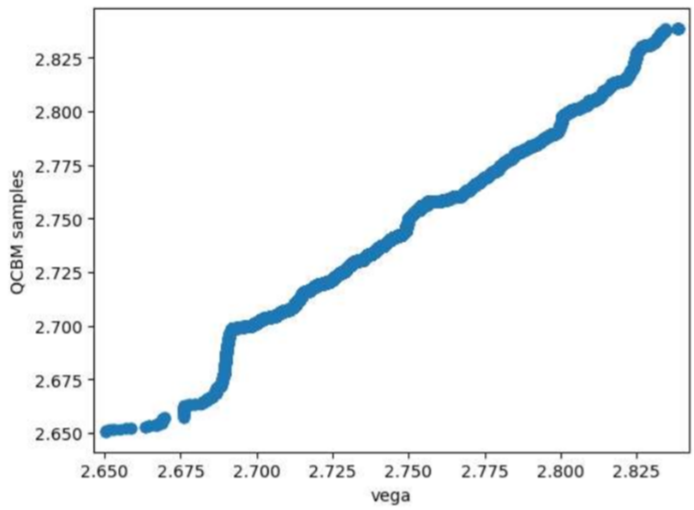
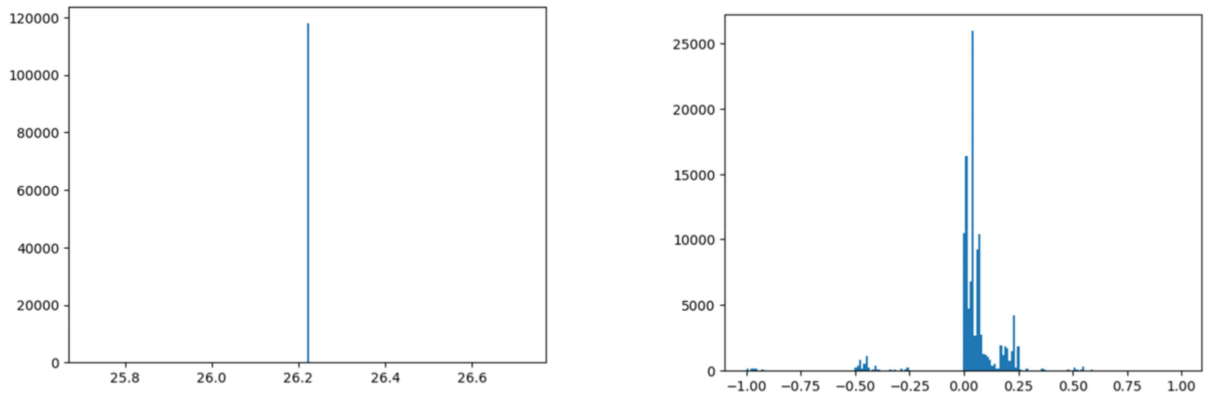


Figure 4: Quantile plot for American option vega



(a) Histogram of training data

(b) Histogram of samples observed

Figure 5: Comparison of training data and observed samples

5.5 Quantum Feature Maps

Another method for encoding time within quantum circuits was through the use of quantum feature maps. We can define this to be $\phi : X \to F$ where F is a Hilbert space. This map transforms $x \to |\phi(x)\rangle$ by way of $U_\phi(x)$.
Here we have two parts, the latent variable and time encoding. The latent variable is a random, uniform $z \in [0, 1]$. This is encoded through the use of $R_x(\theta)$ where

$$R_x(\theta) = \begin{bmatrix} \cos(\theta/2) & -i\sin(\theta/2) \\ -i\sin(\theta/2) & \cos(\theta/2) \end{bmatrix} \tag{5.5}$$

Time in encoded through the use of $R_y(\theta)$ where

$$R_y(\theta) = \begin{bmatrix} \cos(\theta/2) & -\sin(\theta/2) \\ \sin(\theta/2) & \cos(\theta/2) \end{bmatrix} \tag{5.6}$$

It goes that by learning the quantile function of a SDE, we can evaluate it at random uniform z's as t progresses to give a full time-series that obeys the SDE. To do so we

require a circuit that maps z to a sample $G(z) = Q(z)$.
I have taken the example of the Ornstein-Uhlenbeck process to provide a proof of concept. We can define OU to be:

$$dX_t = \theta(\mu - X_t)\,dt + \sigma\,dW_t \tag{5.7}$$

where θ is the mean reversion, μ is the long-term mean, σ is the volatility, and W_t is the Weiner process.
By way of Fokker-Plank we can get the following for the quantile function:

$$\frac{\partial Q(z,t)}{\partial t} = f(Q,t) - \frac{1}{2}\frac{\partial g^2(Q,t)}{\partial Q} + \frac{g^2(Q,t)}{2}(\frac{\partial Q}{\partial z})^{-2}\frac{\partial^2 Q}{\partial z^2} \tag{5.8}$$

where $f(Q,t)$ is the drift term and $g(Q,t)$ is the diffusion term.
Using this, I was able to run some preliminary tests, gaining the following results:

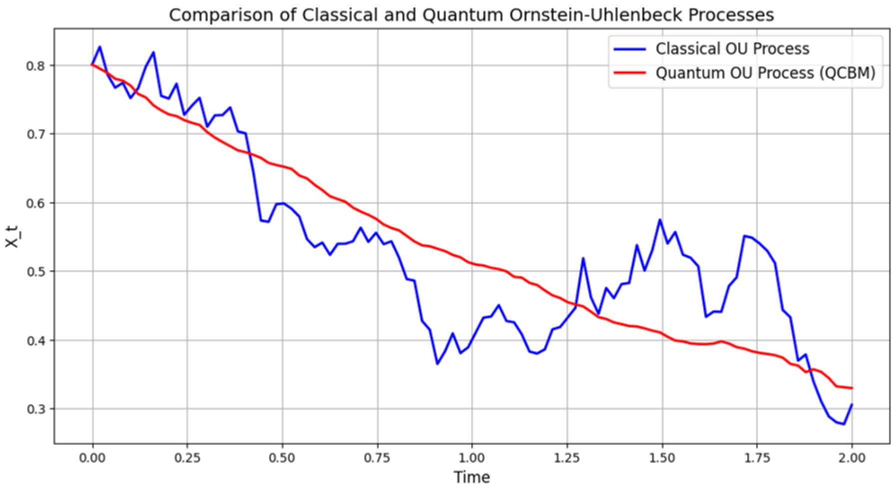


Figure 6: Sampling learnt QF for the OU process

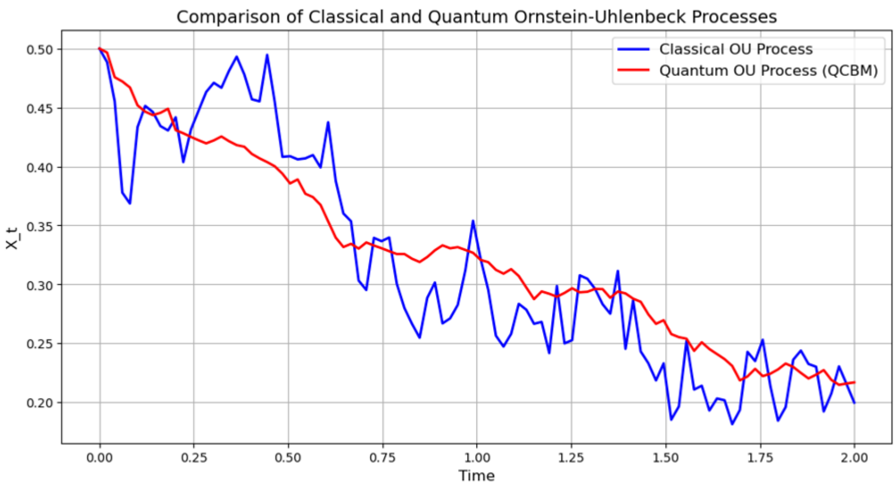


Figure 7: Sampling learnt QF with higher circuit depth

5.5.1 Barren Plateau

A point of concern when searching for the optimal set of θs is the exploration of the large space, here we may observe issues such as barren plateau(BP). BP insists that the gradient of the parameters of a given PQC will vanish exponentially w.r.t the search space. A formal proof can be found in [barren plateau reference].
I will aim to explore using gradient-based methods as well as alternatives such as genetic algorithms.

5.6 Design

Below is the rough proposed high level design with a few components. The market simulation of the underlying will feed into the deep hedger. The trained model will then be compared to a simple Greeks-based hedger on a GBM underlying.

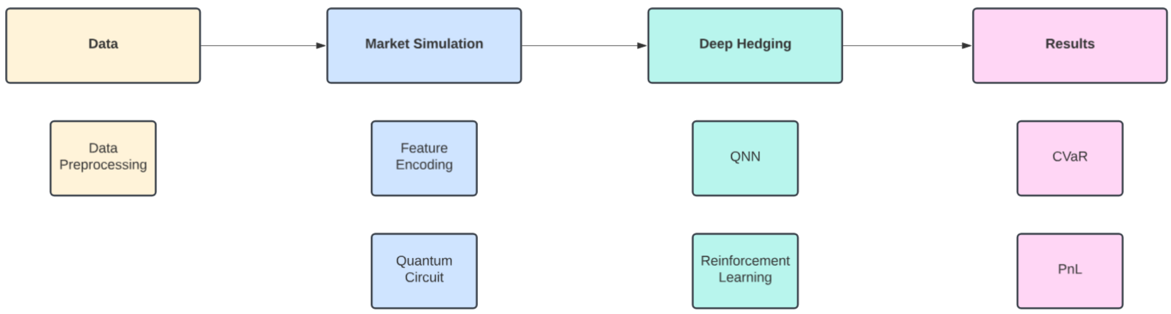
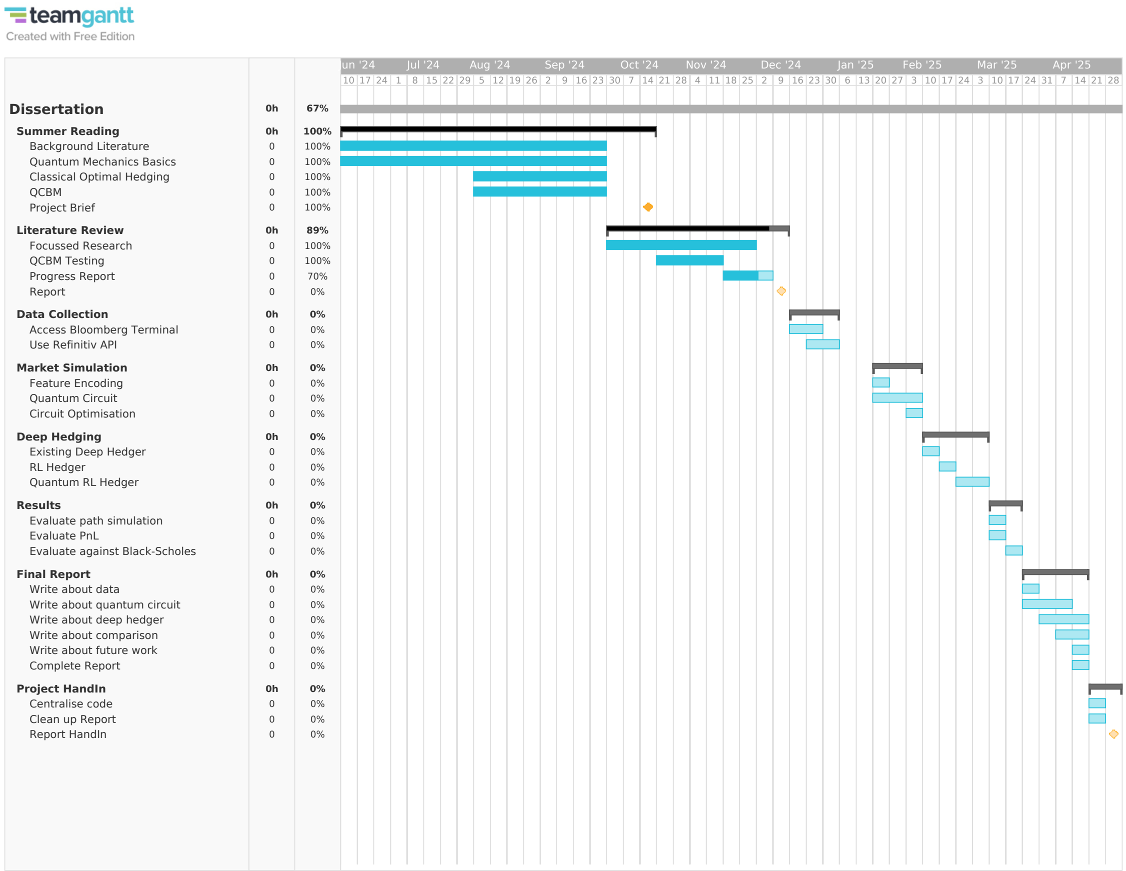


Figure 8: Proposed high-level design

6 Planning

The high level Gantt Chart can be seen in the following figure. I have chosen to split the development into 6 sections, each one with a separate purpose.



7 Risk Assessment

A risk assessment, though pedestrian, is included to be holistic.

Risk	**Probability (1-5)**	**Severity (1-5)**	**Risk Exposure (P x S)**	**Mitigation**
Quantum simulator packages become unavailable	1.5	5	7.5	There are 3 major providers of quantum simulation software: Amazon, Google, and IBM. If one were to become unavailable, I would be able to transfer the quantum circuits to a different provider.
Data becomes unavailable	1	3	3	There are many providers of data for the SX5E and samples have already been downloaded onto a local device. In the worst case, I would be able to run the same code with a different asset.
Illness	4	3	12	All deadlines will be soft deadlines with at least a week of spare time before any hard deadlines. Therefore, hard deadlines will still be met.
Time management issues	3	4	12	The use of time management tools such as Gantt should avoid this.
Project is rendered not possible due to a recent publication	1	5	5	There are many ways to pivot in this project, such as using a different quantum technique.
Device failure	3	5	15	GitHub will be used for source control.

References

[1] Marcello Benedetti et al. "A generative modeling approach for benchmarking and training shallow quantum circuits". en. In: *npj Quantum Inf* 5.1 (May 2019), p. 45. ISSN: 2056-6387. DOI: 10.1038/s41534-019-0157-8. URL: https://www.nature.com/articles/s41534-019-0157-8 (visited on 11/21/2024).

[2] Marcello Benedetti et al. "Erratum: Parameterized quantum circuits as machine learning models (2019 *Quant. Sci. Tech.* **4** 043001)". en. In: *Quantum Sci. Technol.* 5.1 (Dec. 2019), p. 019601. ISSN: 2058-9565. DOI: 10.1088/2058-9565/ab5944. URL: https://iopscience.iop.org/article/10.1088/2058-9565/ab5944 (visited on 11/21/2024).

[3] Nicolas Boursin, Carl Remlinger, and Joseph Mikael. "Deep Generators on Commodity Markets Application to Deep Hedging". en. In: *Risks* 11.1 (Dec. 2022), p. 7. ISSN: 2227-9091. DOI: 10.3390/risks11010007. URL: https://www.mdpi.com/2227-9091/11/1/7 (visited on 11/21/2024).

[4] Hans Buehler, Murray Phillip, and Ben Wood. "Deep Bellman Hedging". en. In: *SSRN Journal* (2022). ISSN: 1556-5068. DOI: 10.2139/ssrn.4151026. URL: https://www.ssrn.com/abstract=4151026 (visited on 11/21/2024).

[5] Hans Buehler et al. "Deep Hedging: Hedging Derivatives Under Generic Market Frictions Using Reinforcement Learning". en. In: *SSRN Journal* (2019). ISSN: 1556-5068. DOI: 10.2139/ssrn.3355706. URL: https://www.ssrn.com/abstract=3355706 (visited on 11/21/2024).

[6] Eliahu Cohen. "What Weak Measurements and Weak Values Really Mean: Reply to Kastner". en. In: *Found Phys* 47.10 (Oct. 2017), pp. 1261–1266. ISSN: 0015-9018, 1572-9516. DOI: 10.1007/s10701-017-0107-2. URL: http://link.springer.com/10.1007/s10701-017-0107-2 (visited on 11/21/2024).

[7] Kalyan Dasgupta and Binoy Paine. *Loading Probability Distributions in a Quantum circuit*. en. arXiv:2208.13372 [quant-ph]. Aug. 2022. URL: http://arxiv.org/abs/2208.13372 (visited on 11/21/2024).

[8] Julien Dudas et al. "Quantum reservoir computing implementation on coherently coupled quantum oscillators". en. In: *npj Quantum Inf* 9.1 (July 2023), p. 64. ISSN: 2056-6387. DOI: 10.1038/s41534-023-00734-4. URL: https://www.nature.com/articles/s41534-023-00734-4 (visited on 11/21/2024).

[9] Giacomo Franceschetto et al. *Harnessing quantum back-action for time-series processing*. en. arXiv:2411.03979 [quant-ph]. Nov. 2024. URL: http://arxiv.org/abs/2411.03979 (visited on 11/21/2024).

[10] Keisuke Fujii and Kohei Nakajima. *Quantum reservoir computing: a reservoir approach toward quantum machine learning on near-term quantum devices*. en. arXiv:2011.04890 [quant-ph]. Nov. 2020. URL: http://arxiv.org/abs/2011.04890 (visited on 11/21/2024).

[11] Santanu Ganguly. "Implementing Quantum Generative Adversarial Network (qGAN) and QCBM in Finance". en. In: ().

[12] Jorge García-Beni et al. "Squeezing as a resource for time series processing in quantum reservoir computing". en. In: *Opt. Express* 32.4 (Feb. 2024). arXiv:2310.07406 [quant-ph], p. 6733. ISSN: 1094-4087. DOI: 10.1364/OE.507684. URL: http://arxiv.org/abs/2310.07406 (visited on 11/21/2024).

[13] L. C. G. Govia et al. "Quantum reservoir computing with a single nonlinear oscillator". en. In: *Phys. Rev. Research* 3.1 (Jan. 2021), p. 013077. ISSN: 2643-1564. DOI: 10.1103/PhysRevResearch.3.013077. URL: https://link.aps.org/doi/10.1103/PhysRevResearch.3.013077 (visited on 11/21/2024).

[14] Iordanis Kerenidis, Jonas Landman, and Natansh Mathur. *Classical and Quantum Algorithms for Orthogonal Neural Networks*. en. arXiv:2106.07198 [quant-ph]. Dec. 2022. URL: http://arxiv.org/abs/2106.07198 (visited on 11/21/2024).

[15] Alex Kondratyev. "Non-Differentiable Leaning of Quantum Circuit Born Machine with Genetic Algorithm". en. In: *Wilmott* 2021.114 (July 2021), pp. 50–61. ISSN: 1540-6962, 1541-8286. DOI: 10.1002/wilm.10943. URL: https://onlinelibrary.wiley.com/doi/10.1002/wilm.10943 (visited on 11/21/2024).

[16] Jin-Guo Liu and Lei Wang. "Differentiable Learning of Quantum Circuit Born Machine". en. In: *Phys. Rev. A* 98.6 (Dec. 2018). arXiv:1804.04168 [quant-ph], p. 062324. ISSN: 2469-9926, 2469-9934. DOI: 10.1103/PhysRevA.98.062324. URL: http://arxiv.org/abs/1804.04168 (visited on 11/21/2024).

[17] Ali Asghar Movahed and Houshyar Noshad. "Introducing a new approach for modeling a given time series based on attributing any random variation to a jump event: jump-jump modeling". en. In: *Sci Rep* 14.1 (Jan. 2024), p. 1234. ISSN: 2045-2322. DOI: 10.1038/s41598-024-51863-5. URL: https://www.nature.com/articles/s41598-024-51863-5 (visited on 11/21/2024).

[18] Pere Mujal et al. "Time-series quantum reservoir computing with weak and projective measurements". en. In: *npj Quantum Inf* 9.1 (Feb. 2023), p. 16. ISSN: 2056-6387. DOI: 10.1038/s41534-023-00682-z. URL: https://www.nature.com/articles/s41534-023-00682-z (visited on 11/21/2024).

[19] Bernt Øksendal. "Application to Mathematical Finance". In: *Stochastic Differential Equations: An Introduction with Applications*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 269–313. ISBN: 978-3-642-14394-6. DOI: 10.1007/978-3-642-14394-6_12. URL: https://doi.org/10.1007/978-3-642-14394-6_12.

[20] Anupama Padha and Anita Sahoo. "Quantum deep neural networks for time series analysis". en. In: *Quantum Inf Process* 23.6 (May 2024), p. 205. ISSN: 1573-1332. DOI: 10.1007/s11128-024-04404-y. URL: https://link.springer.com/10.1007/s11128-024-04404-y (visited on 11/21/2024).

[21] Annie E. Paine, Vincent E. Elfving, and Oleksandr Kyriienko. "Quantum Quantile Mechanics: Solving Stochastic Differential Equations for Generating Time-Series". en. In: *Adv Quantum Tech* 6.10 (Oct. 2023). arXiv:2108.03190 [quant-ph], p. 2300065. ISSN: 2511-9044, 2511-9044. DOI: 10.1002/qute.202300065. URL: http://arxiv.org/abs/2108.03190 (visited on 11/21/2024).

[22] Obdulia Pichardo Lagunas, Juan Martínez-Miranda, and Bella Martínez Seis, eds. *Advances in Computational Intelligence: 21st Mexican International Conference on Artificial Intelligence, MICAI 2022, Monterrey, Mexico, October 24–29, 2022, Proceedings, Part I*. en. Vol. 13612. Lecture Notes in Computer Science. Cham: Springer Nature Switzerland, 2022. ISBN: 978-3-031-19492-4 978-3-031-19493-1. DOI: 10.1007/978-3-031-19493-1. URL: https://link.springer.com/10.1007/978-3-031-19493-1 (visited on 11/21/2024).

[23] Lupei Qin, Wei Feng, and Xin-Qi Li. "Simple understanding of quantum weak values". en. In: *Sci Rep* 6.1 (Feb. 2016), p. 20286. ISSN: 2045-2322. DOI: 10.1038/srep20286. URL: https://www.nature.com/articles/srep20286 (visited on 11/21/2024).

[24] Ali Shaib et al. "Efficient noise mitigation technique for quantum computing". en. In: *Sci Rep* 13.1 (Mar. 2023), p. 3912. ISSN: 2045-2322. DOI: 10.1038/s41598-023-30510-5. URL: https://www.nature.com/articles/s41598-023-30510-5 (visited on 11/21/2024).

[25] Apimuk Sornsaeng, Ninnat Dangniam, and Thiparat Chotibut. "Quantum next generation reservoir computing: an efficient quantum algorithm for forecasting quantum dynamics". en. In: *Quantum Mach. Intell.* 6.2 (Dec. 2024), p. 57. ISSN: 2524-4906, 2524-4914. DOI: 10.1007/s42484-024-00188-7. URL: https://link.springer.com/10.1007/s42484-024-00188-7 (visited on 11/21/2024).

[26] Nikitas Stamatopoulos and William J. Zeng. "Derivative Pricing using Quantum Signal Processing". en. In: *Quantum* 8 (Apr. 2024). arXiv:2307.14310 [quant-ph], p. 1322. ISSN: 2521-327X. DOI: 10.22331/q-2024-04-30-1322. URL: http://arxiv.org/abs/2307.14310 (visited on 11/21/2024).

[27] Magnus Wiese et al. "Deep Hedging: Learning to Simulate Equity Option Markets". en. In: *SSRN Journal* (2019). ISSN: 1556-5068. DOI: 10.2139/ssrn.3470756. URL: https://www.ssrn.com/abstract=3470756 (visited on 11/21/2024).

[28] Neil A. Wilmot and Charles F. Mason. "Jump Processes in the Market for Crude Oil". en. In: *The Energy Journal* 34.1 (Jan. 2013), pp. 33–48. ISSN: 0195-6574, 1944-9089. DOI: 10.5547/01956574.34.1.2. URL: https://journals.sagepub.com/doi/10.5547/01956574.34.1.2 (visited on 11/21/2024).

[29] Xiao-Ming Zhang, Tongyang Li, and Xiao Yuan. "Quantum State Preparation with Optimal Circuit Depth: Implementations and Applications". en. In: *Phys. Rev. Lett.* 129.23 (Nov. 2022). arXiv:2201.11495 [quant-ph], p. 230504. ISSN: 0031-9007, 1079-7114. DOI: 10.1103/PhysRevLett.129.230504. URL: http://arxiv.org/abs/2201.11495 (visited on 11/21/2024).

[30] Christa Zoufal, Aurélien Lucchi, and Stefan Woerner. "Quantum Generative Adversarial Networks for learning and loading random distributions". en. In: *npj Quantum Inf* 5.1 (Nov. 2019), p. 103. ISSN: 2056-6387. DOI: 10.1038/s41534-019-0223-2. URL: https://www.nature.com/articles/s41534-019-0223-2 (visited on 11/21/2024).