Summary of Quantum Deep Hedging by Cherrat et al.
Evan Gray

This paper discusses a quantum approach to hedging, the practice of taking offsetting positions to reduce risk from adverse market movements, especially in derivative trading. Classical hedging strategies, such as those derived from the Black-Scholes model, work well in idealized frictionless markets. However, in reality, traders must navigate transaction costs, market illiquidity, slippage, and non-linear payoff structures. These practical considerations render traditional models inadequate. In recent years, this has led to the rise of Deep Hedging, a data-driven approach that uses reinforcement learning to dynamically learn hedging strategies that maximize a risk-adjusted measure of cumulative return. In the Deep Hedging framework, the financial environment is formulated as a Markov Decision Process, and strategies are learned using methods such as policy search or actor-critic algorithms. The objective is not just maximizing expected return but doing so under risk-aversion constraints, often using utility functions such as the exponential utility, which reflects an investor's tolerance for risk.

Alongside advances in algorithmic trading and strategy using machine learning, quantum computing has emerged as a potential tool for computational finance. Particularly, quantum machine learning aims to enhance learning algorithms by using parameterized quantum circuits as a basis for modeling complex functions. Quantum models can encode and process information in exponentially large Hilbert spaces, potentially offering more expressive models with fewer parameters. However, a key challenge with quantum neural networks (QNNs) has been trainability, as many variational circuits suffer from the barren plateau problem, where gradients vanish exponentially with circuit size. In other words the model loses the ability to train and therefore make strong predictions. The present work addresses this by designing trainable QNN architectures that are both expressive and robust.

This paper introduces a framework called Quantum Deep Hedging, which merges deep reinforcement learning with quantum computing to solve the hedging problem in realistic market settings. The authors propose two families of QNN architectures: orthogonal layers and compound layers. Orthogonal layers operate in the Hamming-weight-1 subspace using Reconfigurable Beam Splitter (RBS) gates and simulate orthogonal transformations akin to classical neural network layers. These layers support efficient training, are interpretable, and avoid re-orthogonalization overhead seen in classical setups.

$$\mathrm{RBS}_{ij}(\theta) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\theta) & \sin(\theta) & 0 \\ 0 & -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Figure 1: RBS gate matrix notation

Compound layers, on the other hand, act on higher Hamming-weight subspaces using Fermionic Beam Splitter (FBS) gates. These provide richer representational capacity by modeling transformations over exterior algebras. Exterior algebras are mathematical structures that represent antisymmetric combinations of features. They allow the network to capture complex, multi-way interactions without redundancy, making them well-suited for identifying intricate patterns in financial data.

$$\mathrm{FBS}_{ij}(\theta) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\theta) & (-1)^{f(i,j,S)} \sin(\theta) & 0 \\ 0 & (-1)^{f(i,j,S)+1} \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Figure 2: FBS gate matrix notation

The authors integrate these quantum architectures into two reinforcement learning algorithms tailored for Deep Hedging. First, they adapt policy-search methods using orthogonal QNNs to model the trading policy. These QNNs are tested in classical financial environments where the market is simulated using Geometric Brownian Motion. Classical architectures like LSTMs and Transformers are compared to their quantum counterparts with the same underlying data and objectives. Remarkably, the quantum models match or outperform classical ones while requiring fewer parameters.

Model	Utility	Terminal PnLs			
		Path 1	Path 2	Path 3	Path 4
LSTM (Classical)	−2.173	−2.578	−1.225	−1.420	−2.671
LSTM (Butterfly – Simulation)	−2.176	−2.586	−1.194	−1.439	−2.671
LSTM (Butterfly – Hardware)	−2.194	−2.610	−1.284	−1.488	−2.658
Transformer (Classical)	−2.167	−2.563	−1.219	−1.411	−2.673
Transformer (Butterfly – Simulation)	−2.195	−2.639	−1.242	−1.388	−2.672
Transformer (Butterfly – Hardware)	−2.539	−3.341	−1.355	−1.247	−2.713

Figure 3: Results of orthogonal QNN experiments on quantum hardware

Second, the authors propose a quantum-native distributional actor-critic algorithm using compound layers. This model not only learns the expected return but also the entire return distribution, which is more aligned with the real-world understanding of risk. Rather than learn a single expected value, this approach provides a more holistic understanding of the distribution of value.

The authors also deploy and experiment on real quantum hardware. The team implemented their models on Quantinuum's H1 trapped-ion quantum processors, running circuits with up to 16 qubits. Two classes of models were tested: policy-search based architectures with orthogonal layers and the distributional actor-critic models using compound layers. In both cases, the results obtained from hardware were closely aligned with those from noiseless classical simulations, demonstrating that such quantum models are not only theoretical but also viable on current noisy intermediate-scale quantum devices.

Algorithm	Utility	Terminal PnLs							
		Path 1	Path 2	Path 3	Path 4	Path 5	Path 6	Path 7	Path 8
Black Scholes	−4.884	−4.602	−5.373	−4.614	−4.263	−5.173	−5.030	−5.017	−4.962
Expected actor-critic (Simulation)	−3.547	+0.078	−6.204	−0.203	+0.967	−6.768	−3.071	−2.984	−6.689
Expected actor-critic (Hardware)	−3.501	+0.213	−6.666	−0.556	+1.067	−6.895	−2.315	−2.569	−6.556
Distributional actor-critic (Simulation)	−3.309	−1.807	−8.313	−3.803	+1.464	−2.736	−1.934	−2.669	−3.944
Distributional actor-critic (Hardware)	−3.369	−1.802	−8.214	−3.648	+1.367	−2.993	−2.047	−2.803	−4.200

Figure 4: Results of actor-critic model against Black-Scholes strategy

From a theoretical standpoint, the paper also provides provable guarantees on trainability. By analyzing the gradient variance across Hamming-weight subspaces, the authors show that their quantum circuits avoid the vanishing gradient problem, even under Gaussian initialization and common measurement settings. This makes the QNNs practical for training on real devices. Furthermore, the compound layers align naturally with the structure of hedging problems where price paths can be encoded as binary strings, and the number of net movements (captured by Hamming weight) is a key determinant of outcomes. This structural insight underpins the architectural choices and contributes to both performance and interpretability.

In summary, the paper presents a comprehensive and scalable solution to the Deep Hedging problem using quantum-enhanced reinforcement learning. The proposed orthogonal and compound QNN architectures achieve high expressivity with fewer parameters, avoid common pitfalls in quantum training, and are validated on real quantum hardware. These methods are broadly applicable to other RL settings, making Quantum Deep Hedging a powerful example of how quantum computing can address high-impact problems in finance. The work sets the stage for continued integration of quantum methods into real-world decision-making, especially in domains where risk, uncertainty, and sequential strategy are paramount.

Reference:

Cherrat, E. A., Raj, S., Kerenidis, I., Shekhar, A., Wood, B., Dee, J., Chakrabarti, S., Chen, R., Herman, D., Hu, S., Minssen, P., Shaydulin, R., Sun, Y., Yalovetzky, R., & Pistoia, M. (2023). *Quantum deep hedging* (No. arXiv:2303.16585). arXiv. https://doi.org/10.48550/arXiv.2303.16585