**Team Roles & Responsibilities – Deep Hedging with Reinforcement Learning Project**

This is just a general outline of what needs to be done. Like a roles type thing. Anyone can do anything.

## 1. Project Lead / Quant Strategist (Travon, ayeee)

* Define project scope, financial motivation, and research questions.
* Decide on reward functions (variance, CVaR, cost-adjusted risk).
* Set the equity universe (SPX/SPY, sector ETFs).
* Guide calibration from OptionMetrics/Bloomberg (volatility surfaces, spreads).
* Write and oversee the intro, methodology, and interpretation sections of the paper.
* Present findings internally (team updates, management) and externally (publication, conferences).

## 2. Machine Learning Engineer (RL Specialist)

* Implement reinforcement learning algorithms (DDPG, PPO, distributional RL).
* Build and maintain training loop (PyTorch/TF), tune hyperparameters.
* Ensure stability in training (entropy regularization, replay buffers, reward shaping).
* Collaborate with Project Lead to align RL objectives with finance intuition.
* Produce policy visualizations (hedge ratio vs. volatility/time to maturity).

## 3. Market Simulator Developer (Quant Engineer)

* Develop simulation environment for equity options under GBM, Heston, SABR, or jump-diffusion models.
* Incorporate real-world frictions: transaction costs, bid–ask spreads, discrete hedging intervals.
* Calibrate model parameters using WRDS OptionMetrics / Bloomberg / CBOE implied volatility surfaces.
* Stress-test the environment under different volatility regimes and market shocks.
* Document assumptions and limitations of the simulation framework.

## 4. Data Engineer / Analyst

* Extract and clean WRDS OptionMetrics and CRSP data.
* Build historical option chain dataset, filtering zero-bid contracts and aligning with SPX/SPY prices.
* Integrate Bloomberg data for transaction costs, dividends, and yield curves.
* Prepare structured datasets for calibration and historical out-of-sample replays.
* Maintain reproducible pipelines (SQL/Python scripts) for data updates and transformations.

## 5. Risk & Validation Analyst

* Define evaluation metrics: hedging error variance, CVaR, VaR, turnover, and cost efficiency.
* Design robustness checks across different volatility regimes and transaction cost levels.
* Run statistical tests: t-tests, bootstrap comparisons against delta/delta–gamma benchmarks.
* Interpret results in financial and risk management terms (tail loss reduction, cost savings).
* Draft and maintain the Results section of the paper with tables, charts, and figures.

## 6. Research Assistant / Documentation Lead

* Conduct literature review (deep hedging, RL in finance, volatility models).
* Manage references and citations; write literature review section.
* Assist in creating charts, LaTeX/Word formatting, and paper editing.
* Document methodology, experiment setup, and reproducibility details.
* Prepare internal presentation decks and support external-facing materials.

# Appendix: Abbreviations & Definitions

|  |  |
| --- | --- |
| Abbreviation | Definition |
| GBM | Geometric Brownian Motion – a stochastic process commonly used to model stock prices. |
| Heston | A stochastic volatility model where volatility follows its own mean-reverting process. |
| SABR | Stochastic Alpha Beta Rho – a stochastic volatility model often used for derivatives pricing. |
| RL | Reinforcement Learning – a machine learning approach where an agent learns by interacting with an environment. |
| DDPG | Deep Deterministic Policy Gradient – an RL algorithm for continuous action spaces. |
| PPO | Proximal Policy Optimization – a stable and widely used reinforcement learning algorithm. |
| CVaR | Conditional Value at Risk – a tail risk measure capturing expected loss beyond a certain percentile (e.g., 95%). |
| VaR | Value at Risk – the maximum loss expected over a given horizon at a certain confidence level. |
| IV | Implied Volatility – volatility derived from option market prices, reflecting market expectations. |
| SPX | Standard & Poor’s 500 Index – index of 500 large-cap U.S. stocks, widely used in derivatives trading. |
| SPY | SPDR S&P 500 ETF – exchange-traded fund that tracks the SPX index. |
| WRDS | Wharton Research Data Services – a data platform with financial and economic datasets. |
| CRSP | Center for Research in Security Prices – a dataset of historical stock prices and returns. |
| DDM | Delta, Delta–Gamma Hedging – classical hedging strategies based on sensitivities (Greeks). |
| SQL | Structured Query Language – a language for querying and managing databases. |
| TF | TensorFlow – a machine learning framework. |
| PyTorch | An open-source deep learning framework commonly used for RL and neural networks. |

| **Category** | **Library** | **Purpose / Use Case** |
| --- | --- | --- |
| **Core Data & Math** | numpy | Core numerical computing, vectors, matrices. |
|  | pandas | Time series, option chain data cleaning/joins. |
|  | scipy | Optimization, statistics, interpolation. |
|  | statsmodels | Econometrics, regression, time-series diagnostics. |
| **ML & RL** | torch (PyTorch) | Neural networks, deep RL models. |
|  | stable-baselines3 | Prebuilt RL algorithms (DDPG, PPO, SAC). |
|  | gymnasium (Gym) | RL environment interface (wrap simulator). |
|  | optuna / ray.tune | Hyperparameter optimization. |
| **Finance / Options** | QuantLib | Option pricing models (Black–Scholes, Heston, Greeks). |
|  | py\_vollib | Black/Black-Scholes pricing and Greeks. |
|  | arch | GARCH, realized volatility models. |
|  | finmarketpy (optional) | Backtesting strategies. |
| **Data Access** | wrds | Access OptionMetrics, CRSP, Compustat. |
|  | blpapi | Bloomberg API interface (historical options/IV). |
|  | xbbg | Python wrapper for Bloomberg pulls. |
|  | yfinance / pandas-datareader | Quick equity/index data for checks. |
| **Simulation** | sdeint | Solvers for stochastic differential equations. |
|  | torchsde (advanced) | Neural SDE models (frontier research). |
|  | numba / jax | Speed up Monte Carlo simulations (JIT). |
| **Visualization & Reporting** | matplotlib | Basic plots. |
|  | seaborn | Statistical visualizations. |
|  | plotly | Interactive charts (e.g., vol surfaces). |
|  | mlflow / wandb | Experiment tracking and logging. |
| **Collaboration** | pytest | Unit testing and reproducibility checks. |
|  | black, flake8, isort | Code formatting, linting. |
|  | sphinx / mkdocs | Documentation generation. |