# Multi-Agent Reinforcement Learning Approaches to Algorithmic Trading with a Hierarchical Agent Structure

COMP0124 Research Project

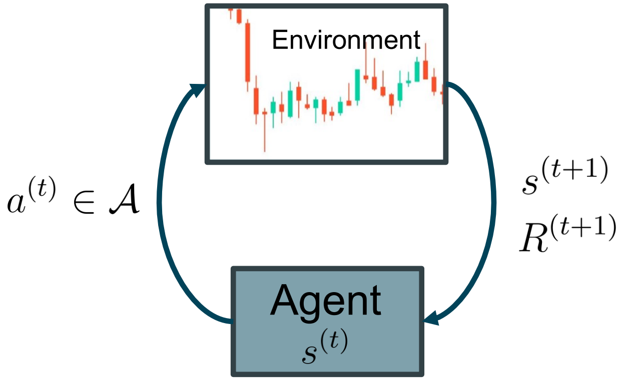

***Group 14***

**Shomit Basu**
**Oli Bridge**
**Isaac Watson**

UCL Department of Computer Science

# Trading as a Markov Decision Process (MDP)

- MDP defined as tuple $\langle \mathcal{S}, \mathcal{A}, \mathbb{P}, R, \gamma \rangle$
  - State space $\mathcal{S}$ - OHLC, current position
  - Action space $\mathcal{A} = \{-1, 0, 1\}$ - sell, hold, buy
  - Transition probabilities $\mathbb{P}$ - from market dynamics
  - Rewards $R(s, a, s')$ - e.g. change in portfolio value
  - Discount factor $\gamma \in [0, 1]$




- Agent wishes to maximise its *discounted return* $G^{(t)} = \sum_{k=0}^{\infty} \gamma^k R^{(t+k+1)}$

- Multi-agent extension: Markov Game $\langle N, \mathcal{S}, \mathcal{A}, \mathbb{P}, R, \gamma \rangle$
  - Each agent has action space $\mathcal{A}_i = \{-1, 0, 1\}$
  - Transition probabilities $\mathbb{P} : \mathcal{S} \times \mathcal{A}_1 \times \ldots \times \mathcal{A}_N \times \mathcal{S} \to [0, 1]$
  - Individual agent rewards $R_i : \mathcal{S} \times \mathcal{A}_1 \times \ldots \times \mathcal{A}_N \to \mathbb{R}$
  - All agents wish to maximise their own return $G_i$

Depend on **all** agents' actions

# Literature Review – Previous Work

- Single-agent trading
  - Various DRL algorithms e.g. DQN[1], DDPG[2], PPO[3] etc. have been implemented
  - Trading applications include single-stock trading, portfolio optimisation, market-making

- Multi-agent trading
  - Hierarchical and *non-episodic* approach to Forex trading introduced by Shavandhi and Khedmati[4]
  - The authors' choice of a non-episodic construct necessitates the use of high latency timeframes with many data points.

[1] M. Taghian, A. Asadi, and R. Safabakhsh, "Learning Financial Asset-Specific Trading Rules via Deep Reinforcement Learning", (2020)
[2] X. Liu *et al*, "Practical Deep Reinforcement Learning Approach for Stock Trading", (2018)
[3] J. Ge *et al*, "Single stock trading with deep reinforcement learning: A comparative study", (2022)
[4] A. Shavandhi, M. Khedmati, "A multi-agent deep reinforcement learning framework for algorithmic trading in financial markets", (2022)

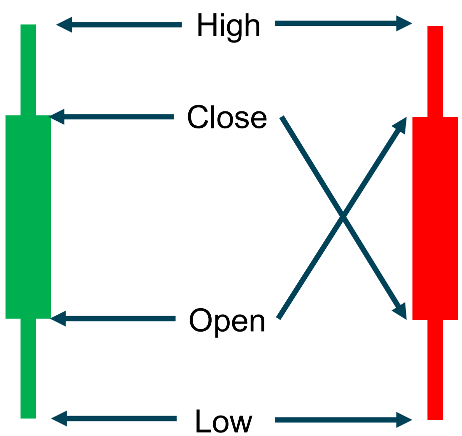
# Research Questions

- Investigate hierarchical multi-agent framework on low-latency trades
  - Can cooperation over multiple longer-term timeframes (daily/weekly etc.) lead to more profitable trading?
  - Is the performance of the multi-agent setting impacted by the choice of the asset?
  - How does the framework perform in different market conditions (bullish, bearish, high volatility)?

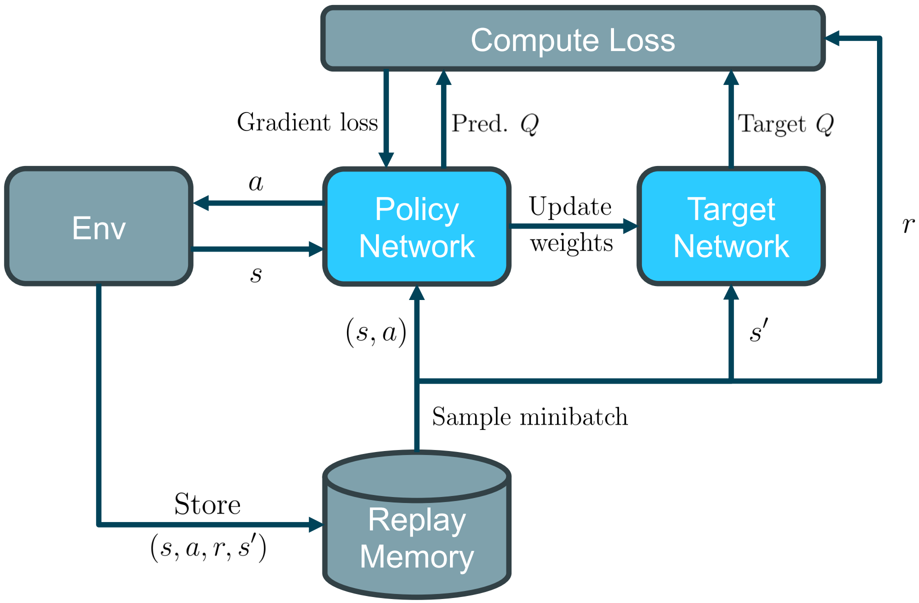
- Episodic framework to make use of smaller dataset sizes
  - Compare single-agent traders with a multi-agent hierarchical framework

# Our Framework – Single Agent RL

- Single agent learns to trade using the Deep Q-Learning (DQN)[5] algorithm with a replay memory buffer
  - DQN policy/target networks are multilayer perceptrons (MLPs)
- Episodic framework
  - Agent learns on training data, and implements its learned behaviour on test data
- Environment iterates through time
  - Provides **states** to the agent (OHLC for a specific timeframe) and **rewards**
  - Receives **actions** from the agent



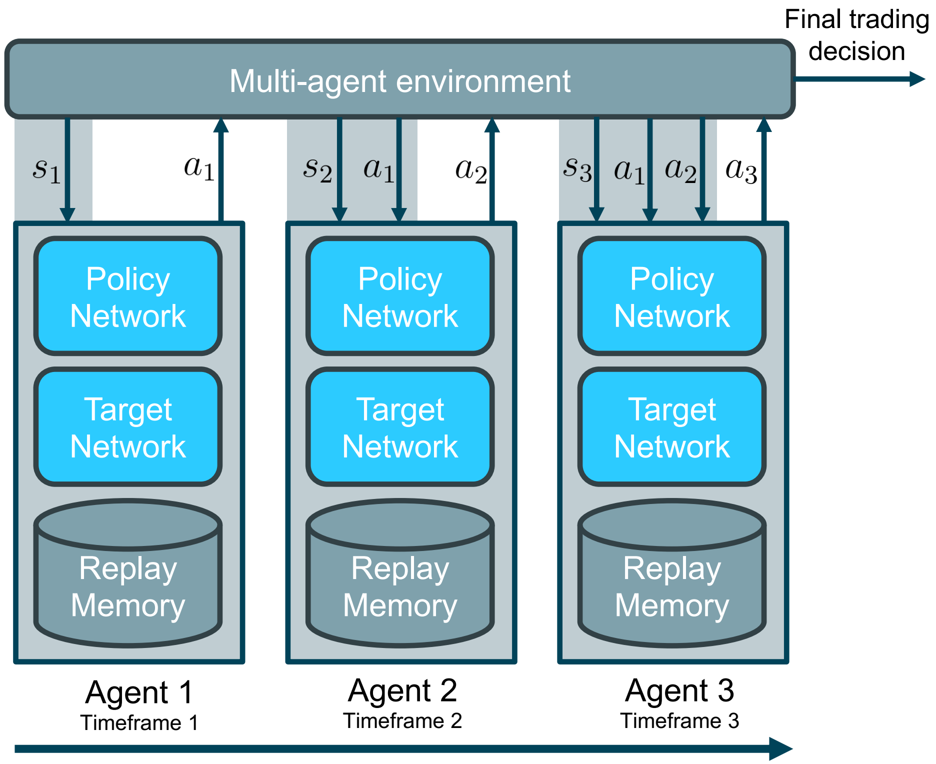

Open, High, Low, Close (OHLC)




[5] V. Mnih *et al.*, “Playing Atari with Deep Reinforcement Learning”, (2013)

# Our Framework – Multi-Agent RL

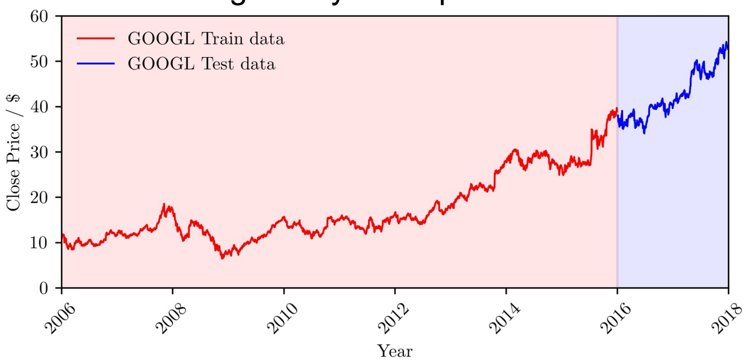
- ❑ Low latency hierarchical cooperation
  - Adapt the single agent trading framework to a *hierarchical agent setting*.
  - Information is passed from lower frequency agents to higher frequency agents.
  - Aim to improve the signal-to-noise ratio for the highest frequency agent.
  - This approach has been effective with high latency FX data.
  - We will look to apply to lower frequency stock data.
- ❑ State space
  - Weekly $s_1 = \{O, H, L, C\}_{t_1}$
  - 3-Daily $s_2 = \{a_1, O, H, L, C\}_{t_2}$
  - Daily $s_3 = \{a_1, a_2, O, H, L, C\}_{t_3}$
- ❑ Action space $\mathcal{A} = \{-1, 0, 1\}$
- ❑ Reward
  - Buy $R_t = \frac{c_{t+n} - c_t}{c_t} \times 100$
  - Sell $R_t = \frac{c_t - c_{t+n}}{c_{t+n}} \times 100$



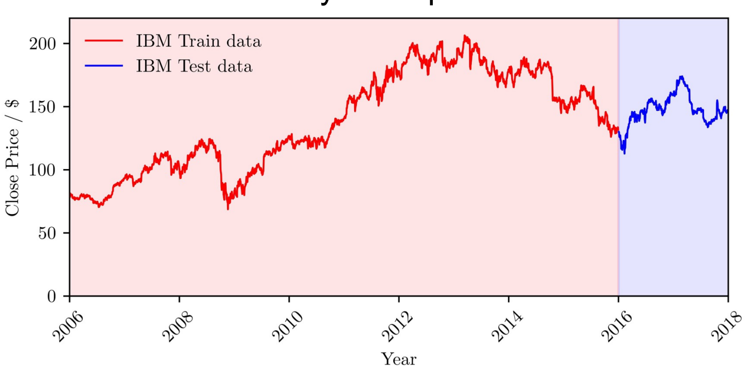
# Preliminary Results – Single Agent Trading

- ❑ Implemented a single stock trading environment with OpenAI Gym API
- ❑ Implemented a DQN single agent that learns to trade single stocks
  - **Daily** OHLC data as input (single timeframe)
  - 10-year training period (Jan 2006 – Jan 2016)
  - 2-year testing period (Jan 2016 – Jan 2018)
- ❑ Bullish stock (Google) and fluctuating stock (IBM)
  - Chosen to see how agent behaves for two differently behaving assets

Google daily close price data



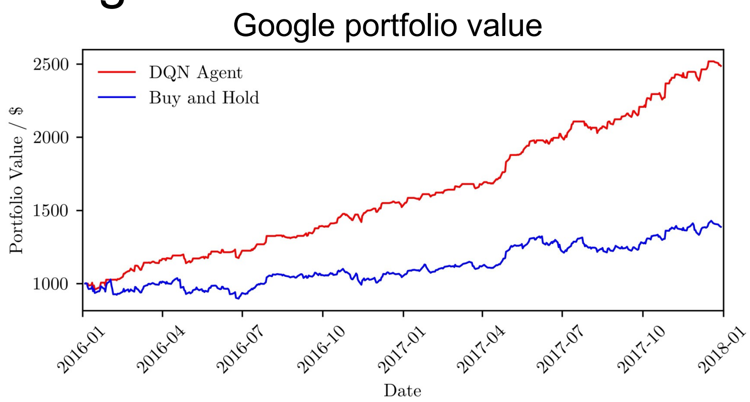

IBM daily close price data

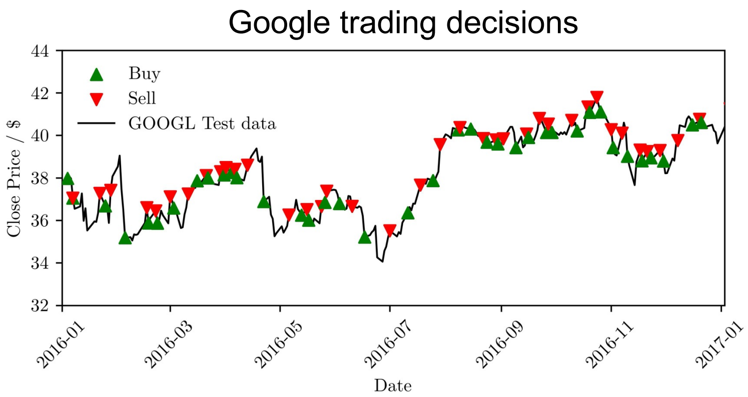


# Preliminary Results – Single Agent Trading

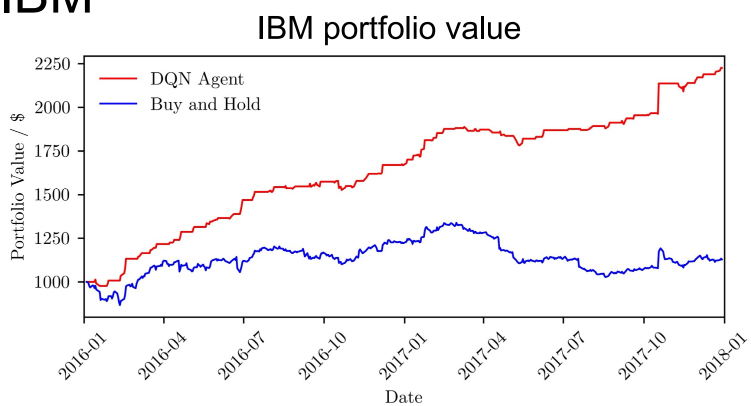
- DQN agent learns to make sensible trading decisions on test data
  - DQN strategy compared to Buy-and-Hold strategy

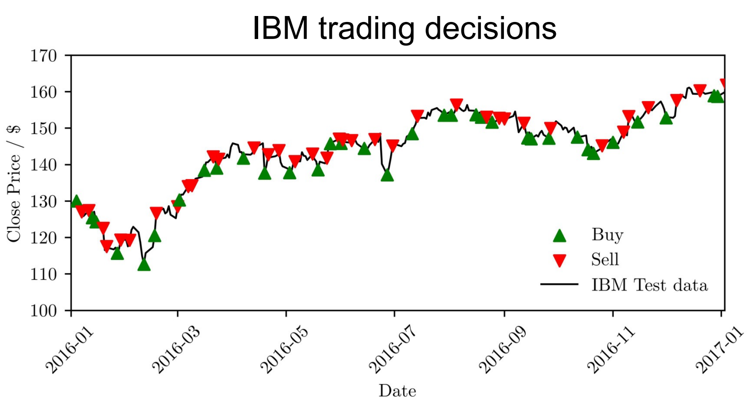
- Google




- IBM



# Project Roadmap

- ❑ Implement a *stock trading environment* for agents to trade on ✓
- ❑ Implement *single trading agent* ✓
- ❑ Utilise more advanced *rule-based baseline strategies* for comparison
- ❑ Extend to *multi-agent framework* with multiple low-latency timeframes
- ❑ Compare the multi-agent trader with individual agent performances

# Thanks for listening!