

Andrea Zappavigna

# Exploration Techniques for a Deep Reinforcement Learning Trading Agent



**Politecnico  
di Torino**

---

# Objective

## **Develop a deep reinforcement learning agent to trade financial securities in any market**

- ▶ Proposed algorithm must reliably outperform traditional financial indexes and other deep reinforcement learning algorithms
  - ▶ It must be able generate positive returns while exposing itself to minimal risk: the key metric for this project is risk-adjusted returns
  - ▶ The agent must learn how to trade under any market trend, including high volatility windows, automatically
-

# Challenges

## **Develop a deep reinforcement learning agent to trade financial securities in any market**

The key difference of our approach is to enhance a single agent to deal with financial markets in their entirety

- ▶ Fair representation of the financial market environment
  - ▶ Experiments must focus on an extended trading window, this is crucial to highlight that the same algorithm can learn effectively in any market trend
-

# Table of contents

00

## Environment

Market framework and  
financial dataset

01

## Agent

Algorithmic approach and  
project details

10

## Results

Backtesting comparison  
with stocks and crypto

11

## Conclusion

Achievements and  
future works

---

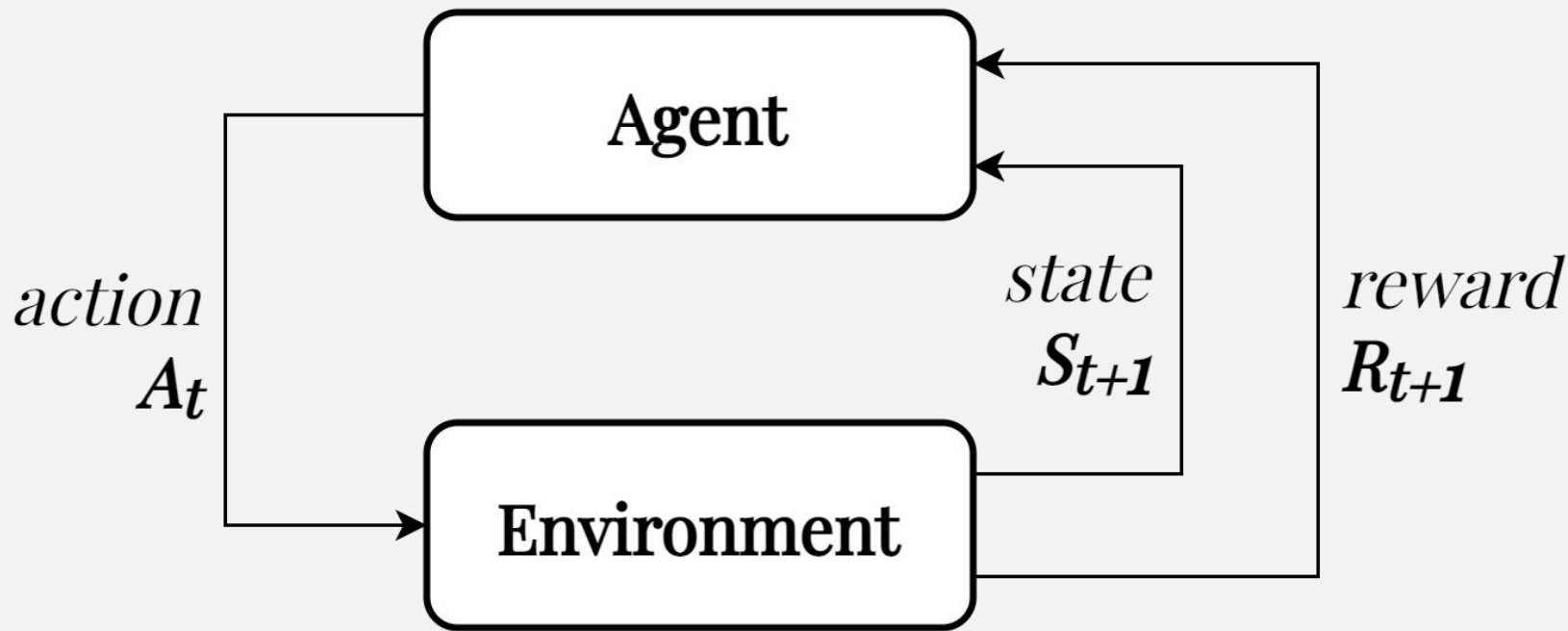
---

00

# Environment

Market framework and financial dataset

# Reinforcement learning problem



---

# Stock trading environment

**OpenAI Gym** environment to mimic a real-world trading exchange:

- ▶ State space: stock information (current prices and owned shares), market features and remaining account balance
- ▶ Action space: actions over the stocks dimension - buy, hold or sell signal at each timestep
- ▶ Reward: difference of total account value (sum of cash balance plus current value of the stock portfolio) between consecutive timesteps

# Market features

```
[date, tic, open, high, low, close, volume,  
macd, boll_ub, boll_lb, rsi_30, cci_30, dx_30,  
close_30_sma, close_60_sma, vix, turbulence]
```

## Technical indicators ●

Help the agent in stock  
forecasting

## ● Market indexes

Help the agent to avoid  
crashes



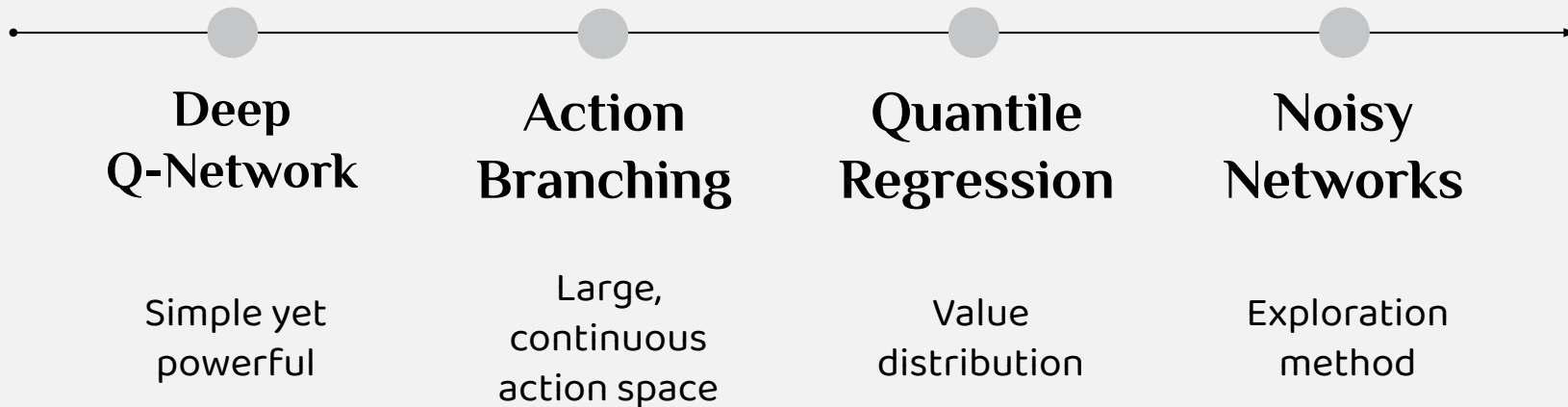
01

# Agent

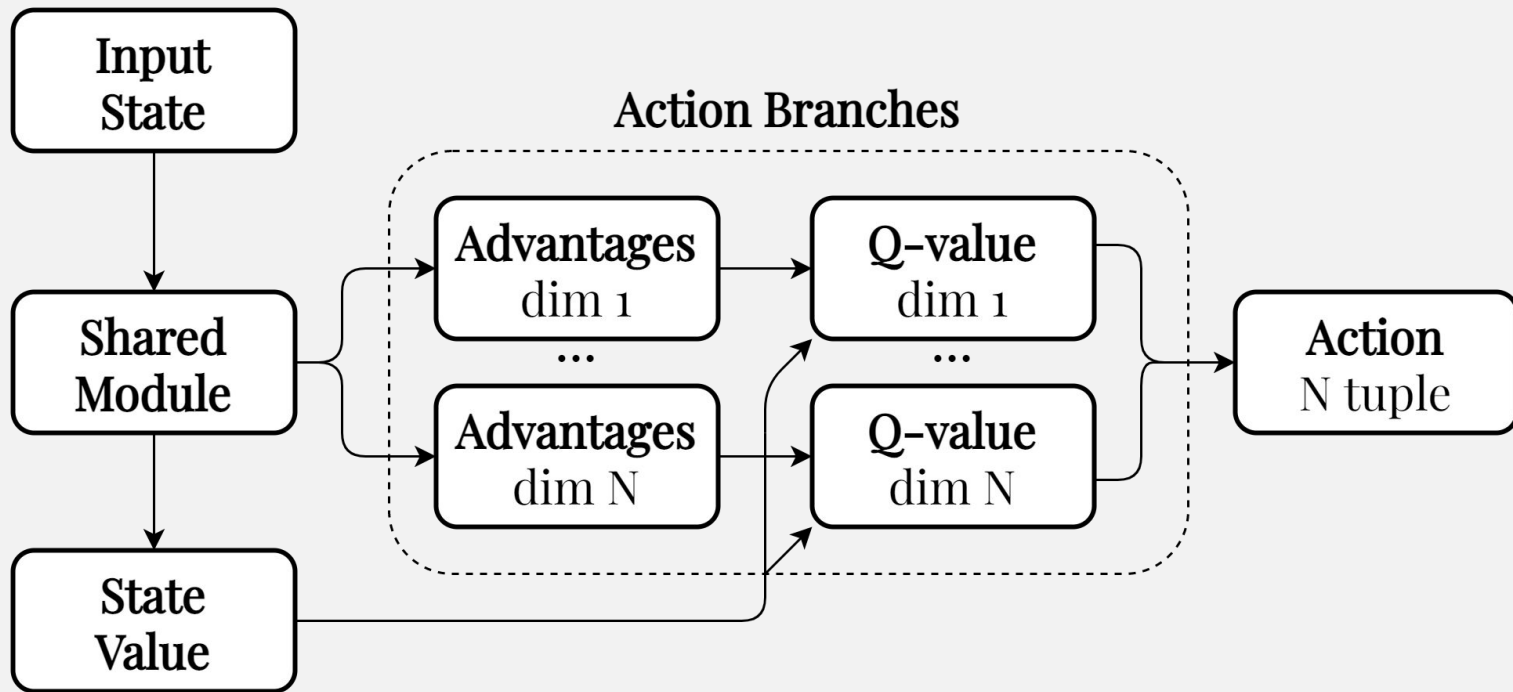
Algorithmic approach and  
project details

---

# TDQN agent



# Action branching architectures



# Distributional reinforcement learning

The agent's goal is to maximize cumulative reward

- ▶ In off-policy agents like DQN, this is achieved by selecting the best action according to the value function:

$$Q_t = \mathbb{E} R_{t+1} + \gamma \cdot \mathbb{E} Q_{t+1}$$

- ▶ In quantile regression we select the best action according to the value distribution instead:

$$Q_t = \mathbb{E} Z_t \rightarrow Z_t = R_{t+1} + \gamma \cdot Z_{t+1}$$

---

# Exploration methods

*Exploration vs Exploitation tradeoff* - how can an agent get as much reward as possible while learning about the environment as quickly as possible?

- ▶ Naïve solution: add random noise by selecting a random action ( $\epsilon - greedy$ )
- ▶ Noisy networks solution: add parameterized noise to the network weights to drive exploration

**As a result the TDQN agent is more resilient  
to the randomness of the environment**

---

---

# 10 Results

Backtesting comparison  
with stocks and crypto

# Stock trading comparison

	TDQN	DDPG	PPO	A2C	DJIA
Cumulative Returns	74.1%	62.5%	57.2%	50.8%	55.9%
Sharpe Ratio	1.01	0.88	0.87	0.73	0.75
Stability	74.5%	59.0%	60.0%	54.8%	67.8%

TDQN strategy manages **30%** improvement over DJIA and **14%** improvement over DDPG in risk-adjusted returns









---

## Yearly returns

Returns	<b>TDQN</b>	<b>DDPG</b>	<b>PPO</b>	<b>A2C</b>	<b>DJIA</b>
2019	23.0%	23.6%	10.6%	18.9%	23.7%
2020	12.2%	7.11%	4.11%	9.96%	5.80%
2021	38.9%	31.8%	42.5%	21.9%	26.4%
tot	74.1%	62.5%	57.2%	50.8%	55.9%

TDQN produces the best **all-round** trading strategy: more reliable (robust) with respect to other deep reinforcement learning agents

# Cryptocurrencies trading

	TDQN	DDPG	EW
Cumulative Returns	99.9%	100%	21.4%
Annual Returns	6.21%	6.21%	1.70%
Sharpe Ratio (vs Volatility)	0.29	0.30	0.19

TDQN strategy manages **43%** improvement over EW in risk-adjusted returns

11

# Conclusion

Achievements and future works

---

# Achievements

## **Develop a deep reinforcement learning agent to trade financial securities in any market**

- ▶ TDQN performs better than state-of-the-art DRL models, obtaining great returns while managing relatively low risk
  - ▶ TDQN can learn effectively even in the highly-volatile cryptocurrency market and obtain solid returns
  - ▶ TDQN is able to adjust to different market trends and implement a robust and reliable trading strategy over an extended trading window
-

# Future works

We are confident that the methods and exploration techniques behind TDQN are great tools for automated trading with reinforcement learning.

- ▶ Integrate these solutions with advanced models like IQN and FQF
- ▶ Expand to even larger state spaces such as S&P 500, STOXX 600

For more implementation details and future updates:



<https://github.com/zappavignandrea>

---

# Thank you for your attention!

Special thank you to my supervisors  
Prof. Luca Cagliero and Dr. Jacopo Fior  
for their guidance in this project

---