MULTIMODAL INVERSE REINFORCEMENT LEARNING ON STOCK TRADING

Lennart Schulze, Tobias Hanl

Final Project – EECS E6892 Reinforcement Learning

May 11, 2023

MOTIVATION

Stock Trading

= Selling and buying assets (shares, options, currencies) on the stock market

Scientific Relevance

- Highly important for the economy and individual businesses
- Observable part of non-observable environment
- Dynamics understood only to limited extent

Learning based approaches

- Predict stock price (regression / DL)
- Predict goodness of trades (classification / DL)
- Make good trades (RL)
- Understand good trading (IRL)

→ Use IRL to understand agents who navigate the unknown system successfully ←

BACKGROUND

[Ziebart 2008]

When reward is difficult to engineer, sparse
When we have expert in complex system

Reinforcement Learning

Given a (partially observable) MDP, find an optimal policy that is expected to generate the maximum reward

Given:

$$M(S, A, P, R, \gamma)$$
$$R:\ S \times A\ \rightarrow \mathcal{R}$$
$$P:\ S \times A\ \rightarrow [0,1]^{|S|}$$

Find:

$$\pi: S \rightarrow [0,1]^{|A|}$$

such that

$$\max_{\pi} E_{\pi}\left[\sum_{t} \gamma^{t} r_{t}\right]$$

Imitation learning (IL)

Given episodes from an optimal expert policy, find a policy that behaves like or learns from the expert

$$M(S, A, P, R, \gamma)$$
$$P: S \times A\ \rightarrow [0,1]^{|S|}$$
$$T = \{t = (s_1, a_1, \ldots, s_n)\}_{\{i=0\}}^{|T|}$$

Inverse Reinforcement Learning (IRL)

Given trajectories from an (sub)optimal expert, recover its reward function (and learn policy)

Find π, r such that

$$E_{T' \sim \pi^{r}}[x(s)] = E_{T^*}[x(s)]$$

Behavioral Cloning

Given trajectories from an optimal expert policy, train policy to match expert (supervised learning)

$$\min_{\pi} d\big(\pi(s), \pi^*(s)\big) \forall s$$

What if the expert is suboptimal?
What if the observation is unseen?
Why does the expert act like this?

…

BACKGROUND

[Xu 2021]

Multimodal Learning

Leverage several data modalities for training and inference of ML models

A) Learn different pieces of information given in different modalities
B) Learn the same information through different modes

Example: Vision-Language Model
Consuming visual and textual information to better answer a question

RELATED WORK

RL for trading

[Liu et al 2018]
[Wu et al 2020]
[Yang et al 2020]

- Proof of concept
- Outperform market average
- Reward has big influence

- → but: no peak performance

Multimodal RL for trading

[Moerland et al 2019]
[Chen et al 2021]
[Daiya et al 2021]

- Increase the observability of the environment
- Enrich the information, get better results

- → but: same shortcomes as RL, harder function approximation

IL & IRL for trading

[Liu et al 2020]
[Dixon et al 2020]
[Park et al 2021]

- Avoid reward engineering
- Mostly behavioral cloning (to guide exploration)

- → but: restricted observation & non-robustness

Novelty: combination of IRL + multimodal for trading

RESEARCH OBJECTIVE

Problem Statement
Observing their merit and recent successes, can we combine IRL and multimodal observations to *understand stock trading* and learn a competitive policy?

Motivation

IRL: Understand instead of repeat

Multimodal: Richer observation, closer to Markov assumption

Research objectives and contributions

1) Demonstrate the technical feasibility of using multimodal state feature vectors in IL methods;
2) evaluate the performance on the complex sample task of profit-optimization on the financial market;
3) benchmark this approach against state-of-the-art unimodal IL, and classical RL on the sample task.

Differences to standard RL setup

- Inverse RL
- Dual ascent (very expensive)
- Pseudo-batch RL
- Partially observable MDP

METHOD: SCIENTIFIC APPROACH

We combine a solution design & experimental approach

Literature Review

Based on theory from literature…
→ Identify algorithms and domain assumptions

Domain mapping

…with tailored mappings to the domain…
→ Translate stock trading task into IRL environment and components

Solution design

…we design a novel solution…
→ Build a model architecture and implement it

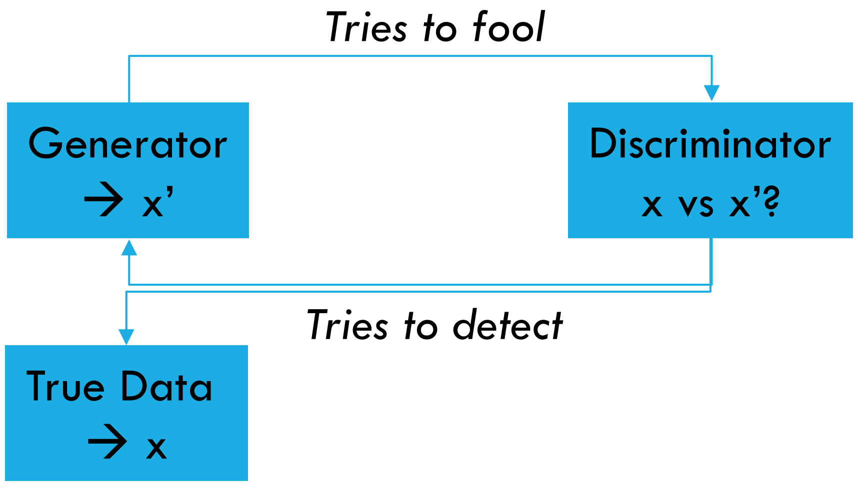
Experiment

…and verify it on real data.
→ Train model and benchmark against other approaches

METHOD: ALGORITHMS

[Goodfellow 2014]

Generative Adversarial Network (GAN)



→ Result: Generator's output indistinguishable from true data

Loss:

$$\arg\min_G \max_D E[\log(D(G(z)) + \log(1 - D(x))]$$

→ Binary cross entropy

→ Minimize loss of generator on best discriminator

Training:

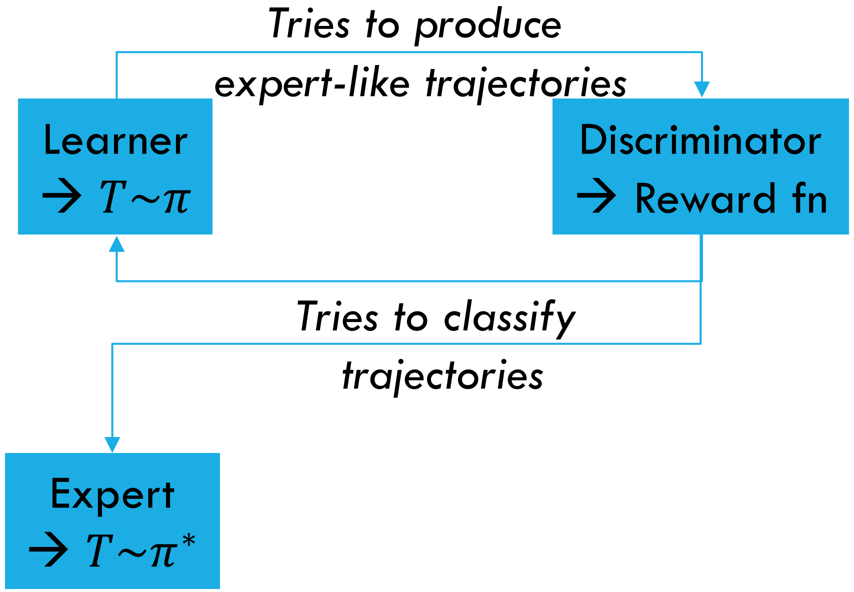
Alternate a gradient step with respect to D's loss

With a gradient step with respect to G's loss

METHOD: ALGORITHMS

[Finn 2016]

GAN in imitation learning



Loss:

$$\arg\min_G \max_D E[\log(D(G(z)) + \log(1 - D(x))]$$

→ Binary cross entropy

→ Minimize loss of generator on best discriminator

Training:

Alternate a gradient step with respect to D's loss
With a gradient step with respect to G's loss

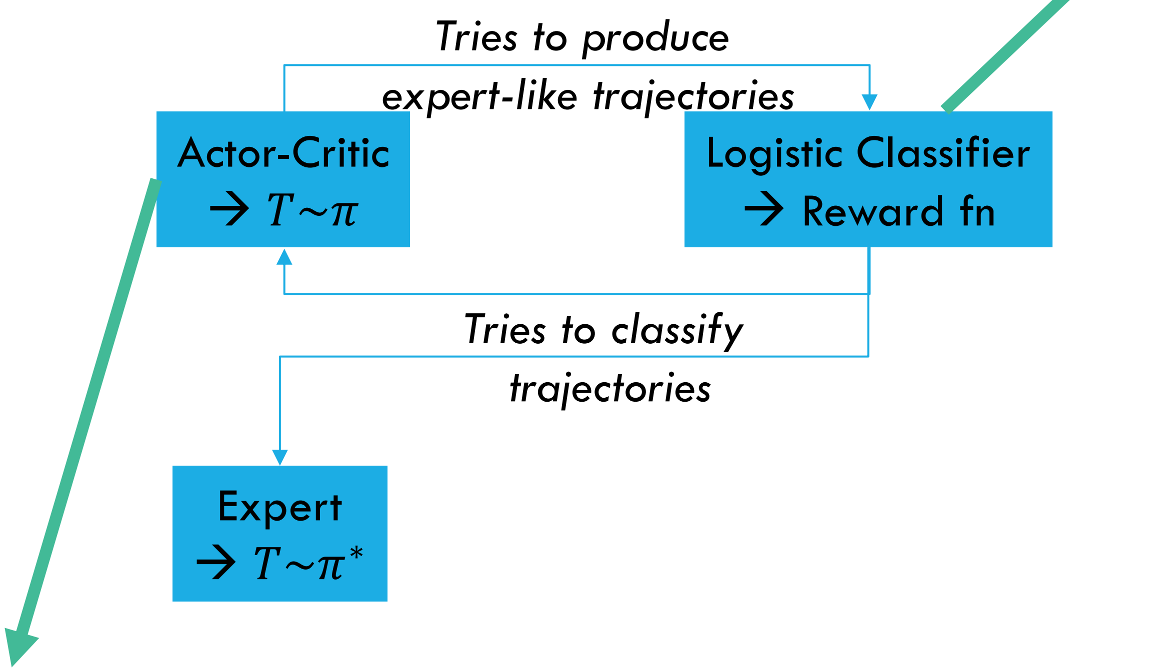
→ Result: Generator's trajectories indistinguishable from expert's trajectories

METHOD: ALGORITHMS

[Fu 2018]
[Schuman 2017]

Reward Net (MLP)
Use multilayer perceptron to nonlinearly map $R: S \times A \rightarrow \mathcal{R}$

Adversarial IRL (AIRL)



Proximal Policy Optimization (PPO)
Policy gradient method using SGD. Objective based on KL penalty. More robust against tuning (step size).

Algorithm

1. Obtain expert trajectories and initialize π, D
2. Iterate until convergence
    1. Collect learner π trajectories
    2. Optimize weights of D wrt to $T \sim \pi$
    3. Update reward function as $r(s,a,s) = \log(D(s,a,s)) - (1 - \log(D(s,a,s))$
    4. Optimize policy wrt updated r using an actor critic method

Benefits

- Supports continuous state space
- Independent of system transition dynamics
- Robust reward, incl. for state-only
- Based on tuple, not trajectory (lower variance)

METHOD: DOMAIN ENGINEERING

Time steps
- Discretize continuous time into days (1 step=1 day)

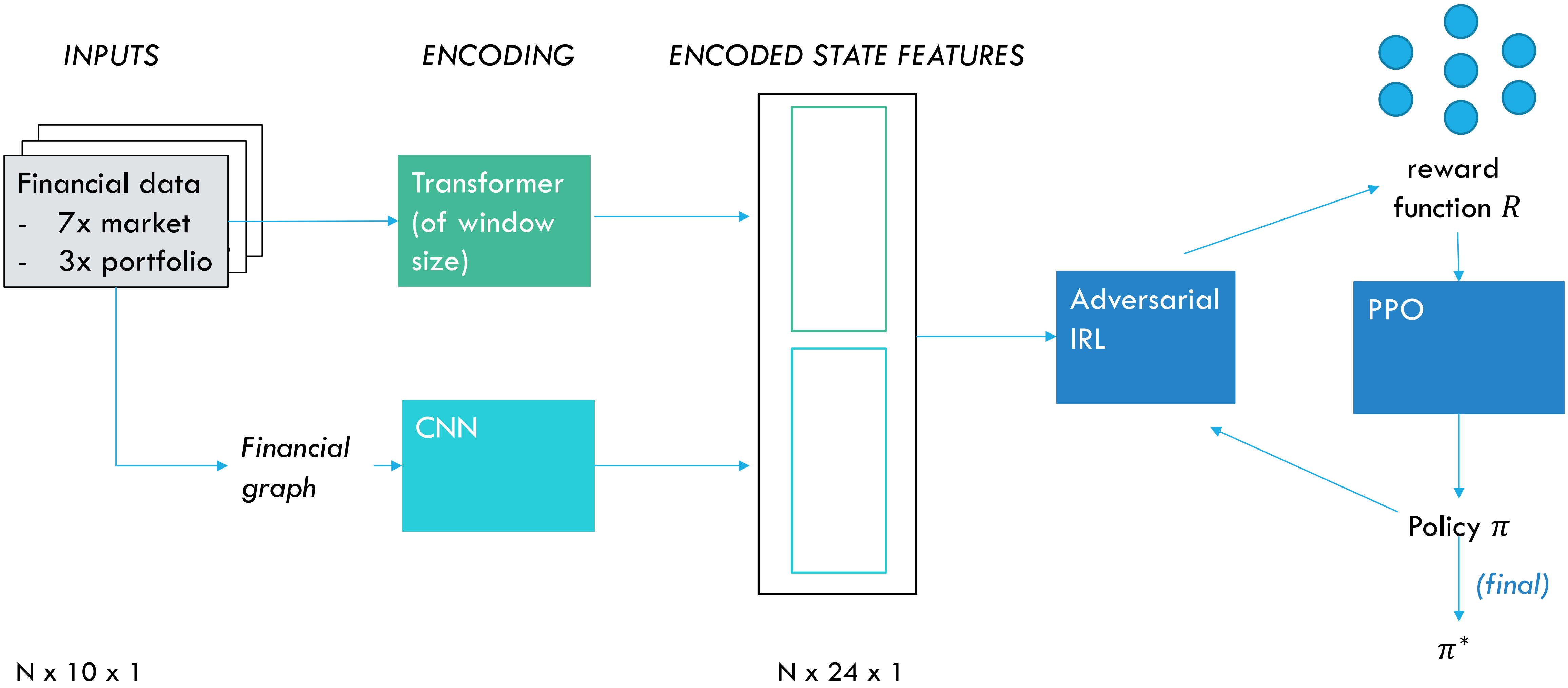
Action space
- {buy, hold, sell}, per step

Observation space: features
- 7x Market (open, high, low, close, volume, #trades, vwap)
- 3x Portfolio (base units, asset units, asset value)

→ Repeated by window size [ws x 10]

(more details in the appendix)

METHOD: SUMMARY



RESULTS: EXPERIMENTAL SETUP

4 setups

- IRL (AIRL + PPO) vs. RL (PPO)
- Base features vs. neural network based encodings

Training environments

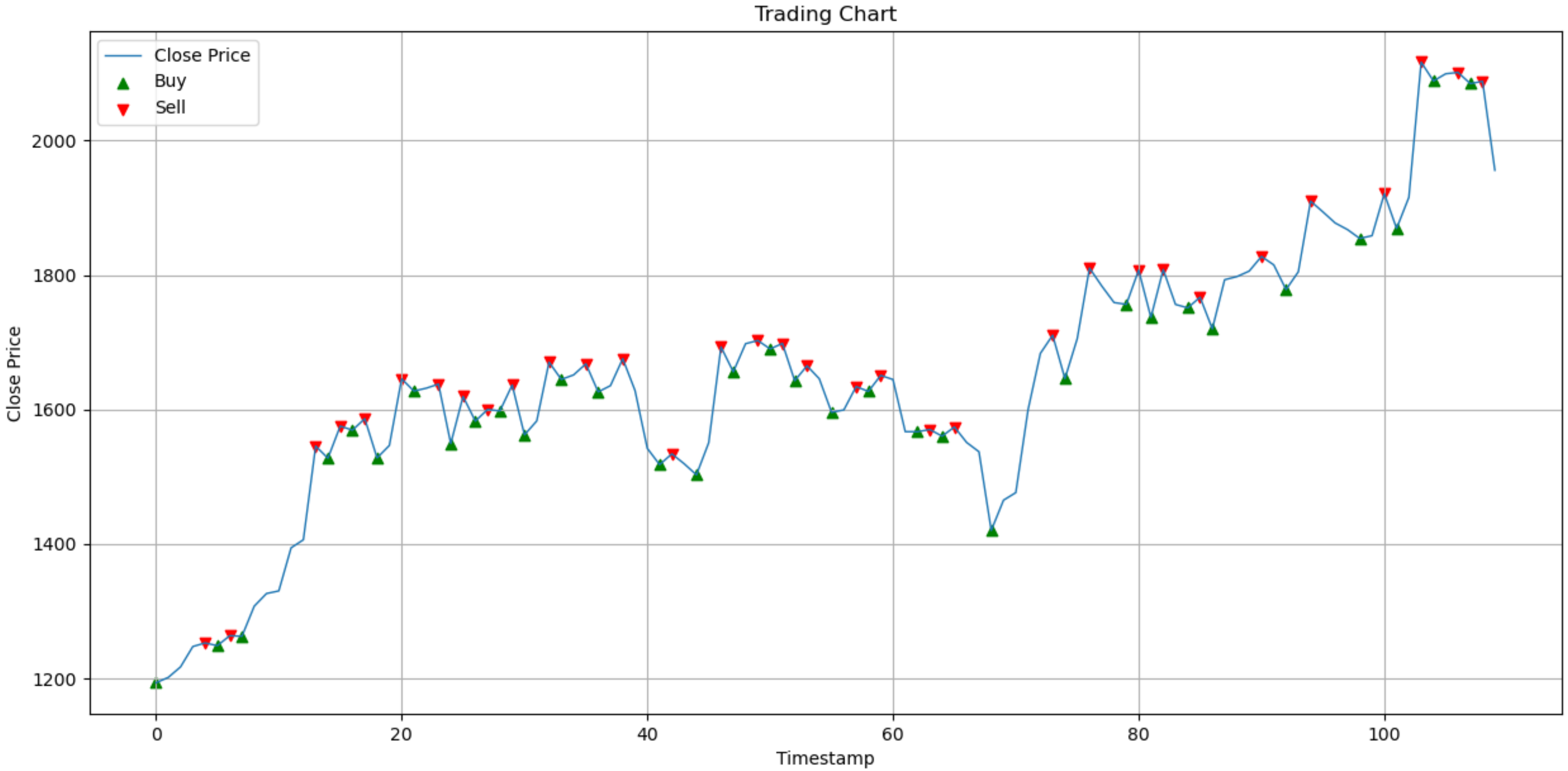
- Randomly sampled historical cryptocurrency exchange data
  - One of BTC/ETH/SOL/ALGO to USD
  - 100-day window between 01/01/2021 – 12/31/2022
- 8 environments for PPO
- 20 environments + expert actions for AIRL
- Number of iterations: 10k
- Simulate environment on historical data by 1) leveraging the fact that individual agents do not affect stock price in the limit (Efficient Market theorem)

Testing environment

- 01/10/2023 – 04/21/2023 with ETH/USD

RESULTS: EXPERT SKYLINE

"buy at every trough, sell at every peak"



RESULTS: TEST DATA

	Profit Before	Profit After
Expert		153.28%

RESULTS: TEST DATA

	Profit Before	**Profit After**
AIRL + PPO + **Encoding**	0.00%	1.49%
PPO + **Encoding**	0.00%	0.00%
Expert		**153.28%**

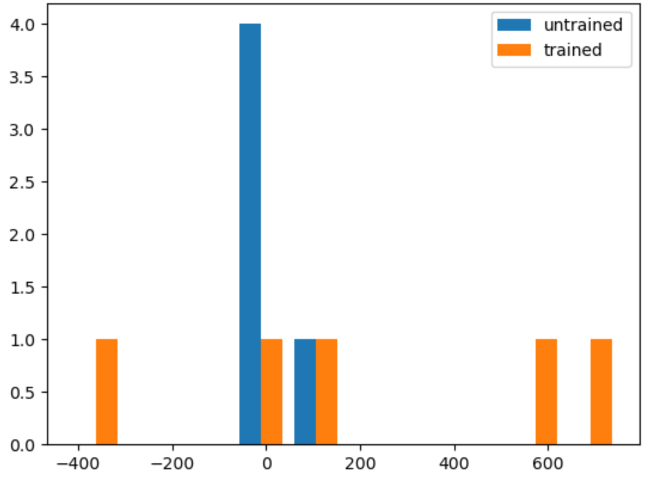
RESULTS: TEST DATA

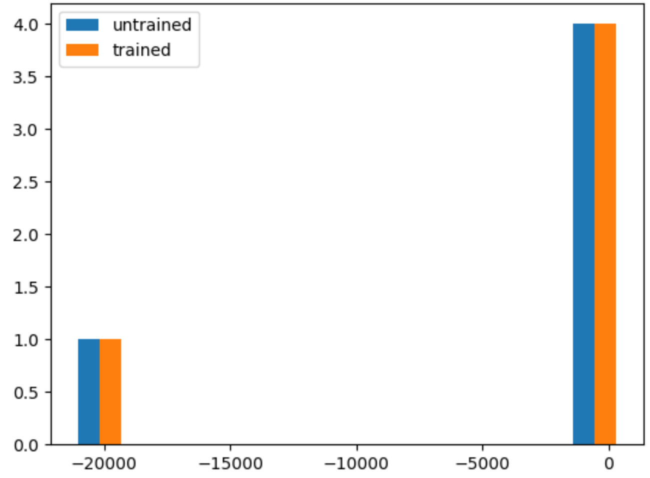
	Profit Before	**Profit After**
AIRL + PPO + **Encoding**	0.00%	1.49%
PPO + **Encoding**	0.00%	0.00%
AIRL + PPO	-3.57%	**31.41%**
PPO	-3.57%	1.98%
Expert		**153.28%**

RESULTS: TRAINING DATA

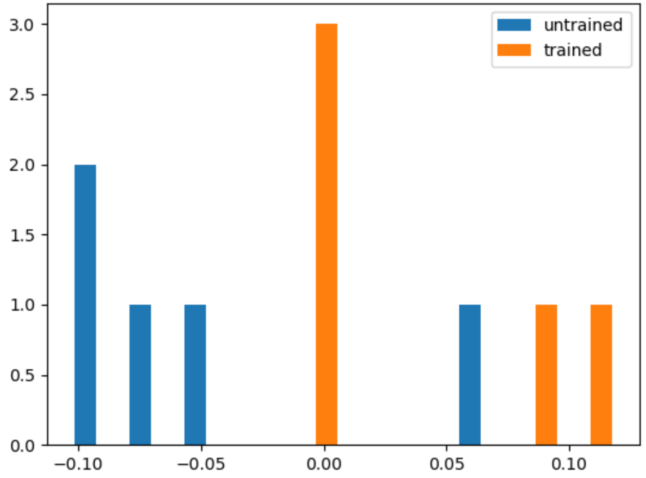
Without Encoding | **With Encoding**

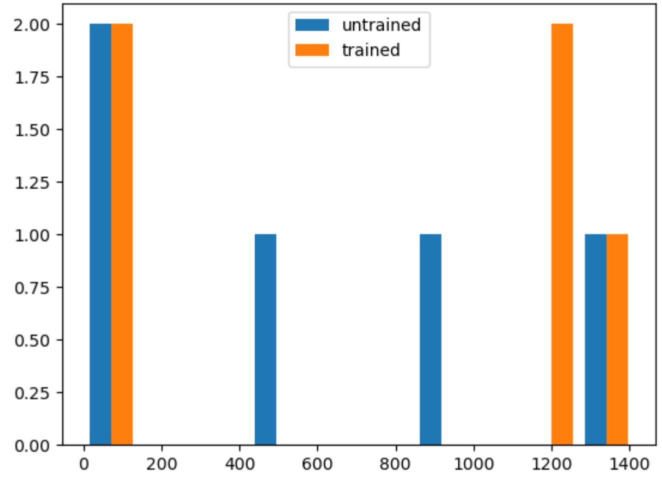
AIRL + PPO





PPO



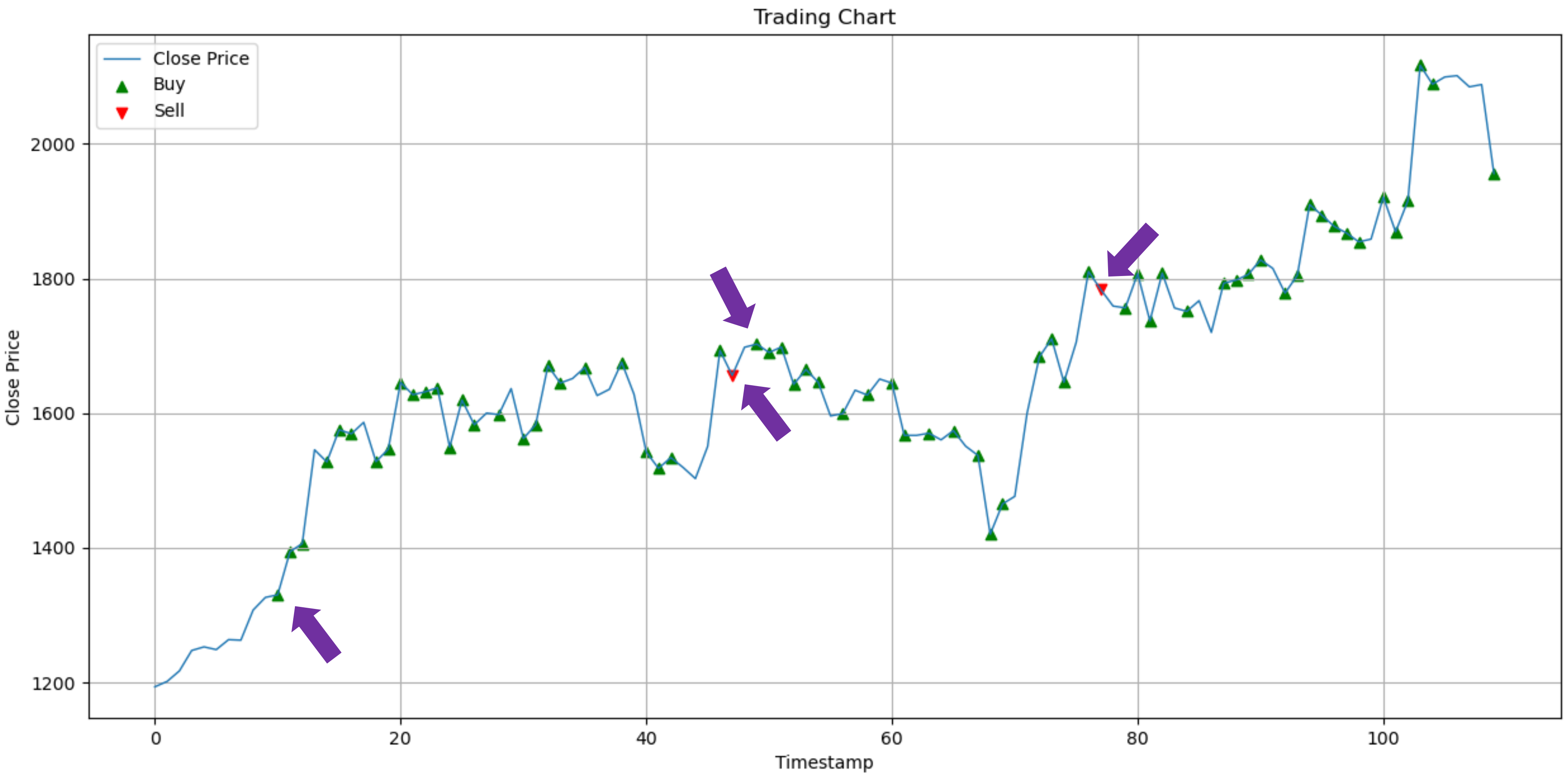


X axis: rewards (engineered only) for benchmark purpose, Profit for

Y axis: frequency (5 environment runs)

RESULTS: TEST DATA

AIRL + PPO - **no** encoding - on test data



CONCLUSION

Answer to the research objectives

1) Demonstrate the technical feasibility of using multimodal state feature vectors in IL methods;

→Feasible, but dual ascent creates training bottleneck

2) Evaluate the performance on the complex sample task of profit-optimization on the financial market;

→ Hard problem, across all approaches rather low performance

3) Benchmark this approach against state-of-the-art unimodal IL, and classical RL on the sample task.

→ Models without multimodal features perform better

→ Imitation learning performs better than reinforcement learning

APPENDIX

METHOD: DOMAIN ENGINEERING

		Rationale
Time steps	- Discretize continuous time into days (1 step=1 day)	
Action space	- {Buy for X USD, hold, sell}, per step	Action needs to performed each step
Feature engineering: features/observations	- Market (open, high, low, close, volume, #trades, vwap) - Agent Portfolio (Cash held, asset units, asset value) → For windows size in steps	Make state dependent on action Previous **data influences** current actions
Feature engineering: encoding	- [Agent Portfolio, Market] → Transformer (sequence=window) - Market → Convolutional NN → 24xwindow component	**History dependence:** Transformer puts attention on meaningful actions **Multimodal:** CNN finds meaningful features, Markov assumption
Expert logic	- In retrospective, sell at every peak, buy at every trough	**Prophet strategy** from literature
Train data episode generation	- Historic data of trades on 4 cryptocurrency symbols against USD	**Price-taker theorem, Efficient Market Hypothesis**