Multiple Pairs Trading for Portfolio Optimization with Reinforcement Learning

Radoslav Georgiev[1]

Supervisor(s): Frans Oliehoek[1]**, Fenghui Yu**[1]

[1]**EEMCS, Delft University of Technology, The Netherlands**

A Thesis Submitted to EEMCS Faculty Delft University of Technology,
In Partial Fulfilment of the Requirements
For the Bachelor of Computer Science and Engineering
June 21, 2026

Name of the student: Radoslav Georgiev
Final project course: CSE3000 Research Project
Thesis committee: Prof. Dr. Frans Oliehoek, Dr. Fenghui Yu, Assoc. Prof. Neil Yorke-Smith

Abstract

Pairs trading has grown increasingly popular over the past several decades, and its application has extended into the domain of portfolio optimization. Reinforcement learning (RL) strategies, particularly Proximal Policy Optimization (PPO), have been used to address this problem. However, while substantial research exists for the single-pair case, a systematic investigation of RL models performing portfolio optimization across multiple pairs simultaneously has been lacking. To address this gap, we develop and compare two PPO models that trade on several cointegrated pairs identified within the energy sector of the S&P 500. The two models differ in their information set: one is given explicit knowledge of the asset pairs it trades, while the other operates without this information, learning to allocate capital from price and portfolio data alone. We find that the pair-aware model achieves an annual return of 20.1% and a Sharpe ratio of 0.877, and maintains consistent performance across varying numbers of traded pairs, though no clear relationship emerges between the number of pairs traded and performance. These results suggest that the multi-pair approach to portfolio optimization might be promising and showcase the need for further investigation.

Keywords: pairs trading, portfolio optimization, reinforcement learning, PPO, multiple pairs, cointegration

1 Introduction

Pairs trading is a popular investment strategy in the financial markets, often utilised by traders and fund managers. The essense of pairs trading relies on the assumption that the two assets in the pair exhibit mean reversion characteristics, which unlock opportunities for profit. This means that when the asset prices diverge, we expect them to converge back to a long-term mean. Profit is realised by selling the high-priced asset and buying the low-priced one.

A pairs trading strategy usually involves the following three steps in order [21][15]:

1. Identifying two or more securities whose prices tend to move together based on historical data.
2. Forming a mean-reverting spread based on the identified securities.
3. Formulating a trading strategy that leverages the constructed spread in order to make a profit.

This framework for exploring arbitrage opputunities, makes pairs trading a desirable approach in solving the portfolio optimization problem [20]. The problem is formulated by Markowitz(1952) as determining the allocation of capital across a set of assets in order to maximise expected return given a risk level. More specifically, we will focus on the case of dynamic portfolio optimization, where the redistribution of capital is performed multiple times within the planning horizon. There has already been considerable work on this formulation of the problem [18][22][8].

In dynamic portfolio optimization, asset weights are continuously redistributed based on market conditions to maximise cumulative returns. This problem maps naturally to the reinforcement learning (RL) framework: the state space comprises stock path information and current portfolio position; the action space defines the redistribution of asset weights; and the reward function is derived from the realised portfolio profit. Consequently, a substantial body of literature has applied RL to this domain and shown promising results. Notable algorithms evaluated in portfolio optimization include Deep Deterministic Policy Gradient (DDPG), Generalized Deterministic Policy Gradient (GDPG), and Proximal Policy Optimization (PPO) [11]. Additionally, utilising pairs trading for portfolio optimization with RL has also been explored [5].

Despite this growing interest in pairs-based portfolio construction, work that allows an RL agent to jointly select trading actions and learn capital allocation weights across multiple pairs remains scarce. Wang et al. [27] uses an approach that trains a DQN agent per pair of assets and then obtains a portfolio optimization strategy by assigning equal weights to the pairs and aggregating the results. Han et al. [10] similarly operate on a single pair at a time: they propose a hierarchical RL framework in which a high-level policy selects one pair from all possible asset combinations and a low-level policy then trades it, jointly optimizing pair selection and trading but never extending the action space beyond the chosen pair. Consequently, while RL has been shown to improve trading decisions within individual pairs and within portfolios of unrelated assets, a systematic investigation of how RL-driven portfolio optimization scales with the number of traded pairs, and whether the allocation across pairs can itself be learned, is, to the best of our knowledge, still missing.

To this end, we develop two RL models for portfolio optimization that trade on several pairs of assets. Both models are given pre-selected pairs of assets and a risk-free bank account on which to distribute weights. The difference is in how much the models know about the pairs and how restricted they are in their trading strategy. The first model, which we call Unstructured, does not know that the assets it is given form pairs and is not restricted in how it distributes the weights between them. The second model, which we call Structured, contains additional information about the pairs in its state, but is restricted to distributing the weights on the spreads of the pairs and not on the individual assets. This additional information and restrictions essentially provide the second model with more structure on how to trade the assets.

The aim of this paper is to compare the two aforementioned RL models for portfolio optimization in terms of the realised profit and see if there is a benefit to the structure given to the second model. We also aim to examine how this comparison changes with a different number of pairs given. Thus, we aim to answer the following two research questions.

RQ1 How does the Structured model for portfolio optimization with asset pairs compare to the Unstructured model

in terms of realised profits?

RQ2 How does the performance of the Structured and Unstructured models change in terms of realised profits with a different number of asset pairs?

The rest of the paper is structured as follows. Section 2 explains the technique for selecting appropriate pairs of assets and describes the implementation of the Unstructured and Structured models. Section 3 provides information on the pair data used and the training of the models. In Section 4, we make a comparison between the two models and examine their behaviour across different numbers of asset pairs. Section 5 discusses and summarises our findings, followed by Section 6, where we assess the paper in terms of responsible research. Finally, Section 7 gives a conclusion to the answers of our research questions and provides recommendations for future work.

2 Methodology

2.1 Pair Selection

The performance of any pairs trading approach is heavily reliant on the quality of the selected pairs. Thus, it is no surprise that multiple pair selection strategies exist, each with its own upsides and downsides.

One approach is the minimum distance method, used heavily in academic literature since the work of Gatev et al. [6]. This approach relies on computing the sum of squared differences(SSD) in order to assess the quality of a pair of assets. A pair of candidates with a lower SSD is preferred over a candidate with a high SSD.

Another approach to pair selection is based on the stationarity of price ratios. This means that the ratio $R_t = P_t^i / P_t^j$ between the two asset prices is modelled as a time series with a constant mean and volatility over time, where deviations from this equilibrium represent trading opportunities. The most common test used to evaluate the stationarity of the ratios is the Augmented Dickey-Fuller (ADF) approach [3]. Pair candidates are ranked based on the statistical significance of the ADF test.

In this paper, we have decided to rely on the cointegration approach for pair selection. Empirical tests among the three approaches suggest that pairs selected via cointegration result in the highest portfolio returns [12].

Cointegration means that two stocks share a long-term equilibrium relationship. Thus, the two assets can be combined to achieve a mean-reverting process. To evaluate cointegration, literature primarily relies on either the Engle-Granger [4] two-step method or the Johansen test [13].

The first step of the Engle-Granger approach relies on an ordinary least squares (OLS) regression to estimate a linear combination of the logarithms of the two asset prices:

$$log(P_{1,t}) - \beta \cdot log(P_{2,t}) = \mu + \epsilon_t, \tag{1}$$

where $P_{1,t}$ and $P_{2,t}$ are the prices of the two assets at time t, β is the cointegration coefficient (hedge ratio), μ is the long-term equilibrium mean, and ϵ_t is the residual error term.

The second step applies the ADF test to the residuals to check for stationarity. While intuitive for a single pair, this method suffers from significant limitations: it is highly sensitive to the choice of the dependent variable (regressing stock i on stock j can yield different results than regressing stock j on stock i), and it is strictly limited to bivariate relationships.

Conversely, the Johansen test overcomes these flaws by utilising a vector autoregressive (VAR) framework to test for cointegration. Instead of assuming a one-way causal relationship, it treats the two assets symmetrically. This advantage suggests that the Johansen test is more suitable for cointegration than the Engle-Granger test [2]. Pair candidates are ranked by the trace statistic given by the test.

2.2 Spread Creation

After suitable assets for pairs are selected via the Johansen test, mean-reverting spreads need to be constructed from these assets. This is done by calculating the hedge ratio β between the logarithms of the two asset prices via Ordinary Least Squares (OLS), yielding the following spread:

$$S_t = \log(P_{1,t}) - \beta \cdot \log(P_{2,t}), \tag{2}$$

where $P_{1,t}$ and $P_{2,t}$ are the prices of the two assets at time t, and β is the OLS coefficient from regressing $\log(P_{1,t})$ on $\log(P_{2,t})$.

The spread can be interpreted as follows: when we buy one unit of the first asset, we need to short sell β units of the second asset, and vice versa.

2.3 PPO Background

The Reinforcement Learning strategy that is used for both the Structured and Unstructured models for portfolio optimization with pairs trading is Proximal Policy Optimization (PPO) [23]. PPO has already been used as one of the models in ensemble strategies [28], but has also shown promising results as a standalone model [14] in the context of portfolio optimization.

PPO is an on-policy reinforcement learning algorithm from the policy gradient family, introduced by Schulman et al. [23] as a simpler and more robust alternative to Trust Region Policy Optimization (TRPO) [25]. PPO is considerably easier to implement and tune while retaining the stability benefits that motivated TRPO.

PPO is an actor-critic method: a policy $\pi_\theta(a_t \mid s_t)$ (actor) and a value function $V_\phi(s_t)$ (critic) are learned jointly, with the critic providing a baseline used to estimate the advantage of each action. The defining feature of PPO is how this advantage is used to update the actor: rather than applying it directly, PPO weighs it by the probability ratio between the new and old policy and clips this ratio to a fixed range $[1-\epsilon, 1+\epsilon]$. This clipping keeps the policy from performing rapid updates.

Three properties of PPO make it particularly well-suited to portfolio optimization in a pairs trading setting:

- **Native support for continuous action spaces.** Allocating capital across pairs requires continuous position sizes rather than a discrete set of actions. PPO parameterises the policy directly and optimizes it via gradient ascent, making it naturally suited to this setting, unlike value-based methods such as DQN [19] that require

action discretisation and scale poorly as the number of pairs grows.

- **Stability under noisy.** Financial returns are volatile and prone to regime shifts, and naive policy gradient updates can be destabilised by a small number of outlier trades. The clipping mechanism addresses this issue by limiting how far the policy can move in response to any single batch of experience.
- **Scalability to high-dimensional action spaces.** Extending portfolio optimization to multiple cointegrated pairs simultaneously requires acting over a multi-dimensional action space. PPO's policy gradient formulation scales naturally to such settings without the combinatorial blow-up faced by discrete-action methods, making it a practical choice for the multi-pair case studied here.

It is worth mentioning that we have also considered and tested another learning strategy - Deep Deterministic Policy Gradient (DDPG) [16] for the implementation of the models, but empirically found slightly poorer performance than that of PPO agents. Another argument against DDPG is the higher complexity of the model as compared to PPO.

2.4 Unstructured model implementation

The Unstructured model operates directly on individual asset prices, with no access to spread- or pair-level information. It trades the same universe of N assets that comprise the pairs selected for the Pairs model, so the two agents differ only in representation and action, not in tradable assets.

We have allowed both models to trade on a risk-free bank account with a 4% annual return. The motivation for this is to avoid forcing the agents into investing in stocks if all stocks exhibit an undesirable behaviour.

State space. For each asset, five features are computed from a 21-day window of log-returns $r_{i,k}$: mean return μ_i (momentum proxy), volatility σ_i (current risk level), skewness s_i (asymmetry between up- and down-moves), 5-day momentum m_i (short-term trend), and a short/long volatility ratio ρ_i (whether risk is currently expanding or contracting).

The choice for these 5 statistics is motivated by examining the related literature and empirical testing. It is worth mentioning that an approach with windows of past log-returns has also been examined, as well as a hybrid approach between past log-returns and the aforementioned statistics.

The 5 features are stacked across assets and concatenated with the agent's current portfolio weights (stocks and cash), giving state dimension of $5N + (N + 1)$.

Action space. The action is an $(N+1)$-dimensional continuous allocation: one weight per asset and one weight for the risk-free bank account. The cash weight $w_{\text{cash}} \in [0, 1]$, while the risky weights $w_i \in [-1, 1]$ are unconstrained in sign, allowing both long and short positions. The sum of all weights is constrained to 1 so that the whole portfolio value is utilised:

$$\sum_{i=1}^{N} w_i + w_{\text{cash}} = 1. \tag{3}$$

Reward. At each step, the portfolio log-return combines the weighted stock and risk-free bank account returns:

$$r_t^p = \sum_{i=1}^{N} w_i \, r_{i,t} + w_{\text{cash}} \, r_f, \tag{4}$$

where $r_{i,t}$ is the log-return of asset i at time t and r_f is the daily risk-free log-return.

Turnover captures how much the allocation changes between consecutive steps:

$$\tau_t = \sum_{i=1}^{N} \left| w_{i,t} - w_{i,t-1} \right| + \left| w_{\text{cash},t} - w_{\text{cash},t-1} \right|. \tag{5}$$

In order to reduce rapid weight changes in the portfolio, we have decided to include a penalty λ_{tc} for the turnover. This penalty can be interpreted as transaction costs. However, we include it only in the environment and do not consider it in the final results.

The reward function is computed by:

$$R_t = r_t^p \, (1 - \lambda_{\text{tc}} \, \tau_t) \tag{6}$$

2.5 Structured model implementation

The only difference between the Structured model and the Unstructured one is the added restriction for trading only on the pair spreads instead of directly on the stocks, and the inclusion of additional pair-specific statistics in the state.

State space. Each cointegrated pair k is represented by its spread return series rather than by the two underlying assets. The four baseline statistics $(\mu_k, \sigma_k, s_k, m_k)$ are recomputed on spread returns, and five mean-reversion-specific features are added: a rolling z-score, its velocity and acceleration, an Ornstein–Uhlenbeck half-life ratio, and a short/long volatility ratio.

The half-life is the expected time for the spread to revert halfway back to its long-run mean after a deviation, under the assumption that it follows a mean-reverting Ornstein-Uhlenbeck process [26]. It is derived from the mean-reversion speed κ as $\text{HL} = \ln(2)/\kappa$; a shorter half-life indicates faster mean reversion. The state includes the ratio of the rolling (test-time) half-life to the half-life estimated on training data, signalling whether reversion has sped up or slowed down relative to the historical regime.

Action and reward. Both are unchanged in form from the Unstructured model, with bets on spreads replacing weights on assets and spread returns replacing asset returns in r_t^p and τ_t.

2.6 Performance Metrics

To evaluate the performance of the Unstructured and Structured models, we rely on the following 5 metrics: end value of the portfolio, annual return, annual volatility, Sharpe Ratio, and maximum drawdown.

The end value of the portfolio is quite self-explanatory - it is the value that our portfolio has at the end of the testing period. The annual return and annual volatility are percentage metrics that are scaled for a year with 252 trading days, which is a common standard in the financial field.

The Sharpe ratio [24] is a common performance metric, which is computed via:

$$\text{Sharpe Ratio} = \frac{r_p - r_f}{\sigma_p}, \tag{7}$$

where r_p is the annual portfolio return, r_f is the annualized risk-free rate, and σ_p is the annual volatility.

We can interpret the Sharpe ratio as the amount of excess annual return we obtain per unit of risk taken - the higher the value, the better.

Finally, the maximum drawdown shows us the largest peak-to-trough decline in portfolio value observed over the testing period, computed via:

$$\text{Max Drawdown} = \min_t \left(\frac{V_t - \max_{\tau \le t} V_\tau}{\max_{\tau \le t} V_\tau} \right), \tag{8}$$

where V_t is the portfolio value at time t.

The maximum drawdown specifically captures the worst-case loss an investor would have experienced had they entered the portfolio at its peak and held through the subsequent decline, making it an intuitive and practical measure of downside risk.

It is worth mentioning that the end value of the portfolio, the annual return, and the Sharpe ratio are closely related metrics. However, the inclusion of all 3 is motivated as they provide information on different aspects of the realised profits of the two models, which is the main criterion we wish to examine in our first research question.

Another thing that should be considered is that all 5 metrics are heavily dependent on the choice of the initial pool of assets. Assets with high returns will most likely lead to portfolio optimization strategies with high returns. Thus, the obtained 5 metrics for the Unstructured and Structured models should not be taken out of the context of the selected stocks.

3 Experimental Setup

3.1 Dataset

To answer our research questions, we select 18 stocks from the energy sector of the S&P 500 via Yahoo Finance. The selected stocks are: XOM, CVX, COP, EOG, SLB, MPC, PSX, VLO, WMB, OXY, HAL, DVN, BKR, FANG, APA, NOV, CTRA, and PR. We use the daily asset paths from 01.01.2015 to 01.01.2022 for train data, and the daily asset paths from 01.01.2022 to 01.01.2024 for test data.

The resulting asset paths have lengths of 1440 and 501 data points, respectively. Although these limited data sizes might provide limitations to the training and testing of the Structured and Unstructured models, we have decided to stick with real historical data both for training and testing. The main argument behind this decision lies in the difficulty of simulating authentic financial data that manages to capture the cointegration relationship between the stocks. One approach that was tried but later dismissed in simulating cointegrated pair data is modeling the spread and the returns as AR(1) processes [17].

3.2 Selected Pairs and Spreads

To select appropriate asset pairs, we conduct the Johansen test at a 95% confidence level on the training asset paths across all 153 pair combinations. The test identifies 33 pairs with sufficient cointegration characteristics, suggesting that the selected asset pool is appropriate.

We rank the 33 pairs based on their trace parameter given by Johansen's test, and select the top 10 pairs to use in our experiments. Note that this results in pairs that might share assets. For example, stock APA is used in 3 distinct pairs. We also calculate the hedge ratio β used for the creation of the spreads of the pairs based on the training asset paths. The 10 selected pairs are shown in Table 1.

Table 1: Johansen cointegration test results for the 10 pairs with the biggest trace statistic, sorted by trace statistic. Hedge ratio β for spread creation is also showcased.

Asset A	Asset B	Hedge Ratio	Trace Statistic
MPC	WMB	1.2810	30.20
BKR	APA	0.5060	28.53
EOG	FANG	0.7393	28.13
WMB	DVN	0.2169	27.64
CVX	WMB	0.5590	26.83
WMB	APA	0.1068	26.49
WMB	FANG	0.2508	26.12
VLO	WMB	0.8364	25.95
WMB	PR	0.0717	25.66
SLB	APA	0.8695	25.40

In Figure 1, we can see the scaled and adjusted asset paths of the first two selected pairs of assets. In both pairs, the second asset is scaled by the hedge ratio β and increased by the intercept α, which is the average of the spread between the two train asset paths. We observe that during the training paths on which the pairs are selected and spreads are formed, the asset paths display a very close relationship to one another. However, as we can see from the pair MPC/WMB, this relationship is not guaranteed during the testing period.

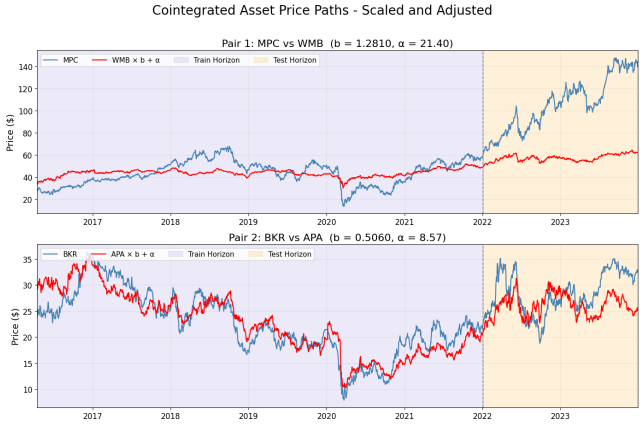

Figure 1: Asset paths of MPC/WMB and BKR/APA pairs during the training and testing periods. The second asset ($Asset\_B$) in each pair is adjusted via $Asset\_B \times b + \alpha$, where b represents the Johansen hedge ratio and α denotes the baseline intercept computed from the training mean spread.

One approach we tried in order to mitigate this effect was

to have the spreads recalculated during training and testing via a rolling β approach. This means that the hedge ratio β is recalculated several times based on a window of past observations. Surprisingly, this approach did not lead to a meaningful improvement in the Structured model, and we therefore removed it for the sake of simplicity.

3.3 Unstructured and Structured Model Training

We have trained and tested the Unstructured and the Structured models on a different number of asset pairs, namely 1, 3, 5, and 10. In each variant, both models have conducted 5 iterations of training and testing, and results have been reported for the averages of those 5 iterations.

Due to a constraint in resources, the hyperparameters have been configured only for the case of 5 pairs of assets, and the same hyperparameters are reused for both models across all variants (1, 3, and 10 pairs included). Therefore, here we will focus on describing the training setup for 5 pairs of assets, as the other cases are analogous.

For 5 pairs, the Structured model operates on the first 5 pairs specified in Table 1 and has a state dimension of 51 and an action dimension of 6. The Unstructured model operates on the 8 assets in those pairs and has a state dimension of 49 and an action dimension of 9. The portfolio value at the start of each training episode and testing iteration is \$10000. The magnitude of this value has no effect on the logic of the models and is simply used for better illustrative purposes.

The values of the hyperparameters used by both models are displayed in Table 2.

Table 2: PPO training hyperparameters shared by the Unstructured and Structured models.

Parameter	Value	Description
N_EPISODES	100	Training episodes
EPISODE_LEN	500	Steps per episode
ACTOR_LR	3e-4	Actor learning rate
CRITIC_LR	3e-3	Critic learning rate
GAMMA	0.99	Discount factor γ
LAM	0.95	GAE parameter λ
CLIP_EPS	0.2	PPO clip range ϵ
K_EPOCHS	4	Epochs per update
VF_COEF	0.5	Value loss weight
ENT_COEF	0.01	Entropy bonus weight
BATCH_SIZE	64	Minibatch size
HIDDEN	128	Hidden layer width
TC_COEF	0.0005	Turnover penalty weight

In order to avoid overfitting on the training data set, in each episode of training, we take a random window of 500 consecutive observations. This window size is motivated by both empirical results during hyperparameter tuning and by the fact that the testing set has a similar length of 501. Since the training data has a size of only 1440, we are aware that the random windows introduce a bias towards the middle of the training set.

Other notable parameters are the learning rates of the actor (3×10^{-4}) and the critic (3×10^{-3}). It is hard to achieve the right balance between the critic and the actor and to prevent one from dominating the other. It is usually the case that the actor's learning rate is lower, since in PPO, the actor's updates directly reshape the policy that collects the data used for updating the model.

To gain a better insight into the training of the Unstructured and the Structured models, Figure 2 shows the learning curves, plotting the average portfolio value at the end of each episode across the 5 training iterations. Both models display an upward trend, suggesting that they manage to learn from the training data. However, the Unstructured model exhibits considerably more variation during training. One explanation for this is the bigger action dimension of the Unstructured model as compared to the Structured one.

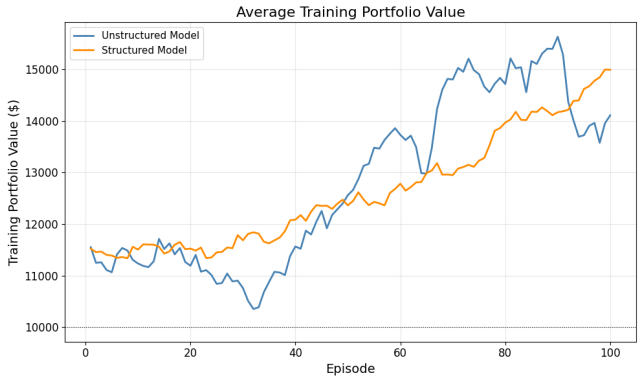

Figure 2: Average portfolio value at the end of the training episodes for the Unstructured and Structured models. Some smoothing has been utilized in order to better visualize trends.

Even though the upward trends in Figure 2 are present up until the end of the training episodes, the number of episodes is kept at 100 as we empirically found that a larger number of episodes leads to considerable overfitting.

4 Results

We compare the Unstructured and the Structured models on 5 asset pairs and take a look at the average results across the training and testing iterations with regard to the unseen test data. Both models are heavily dependent on the overall market movement - a higher market return should lead to higher returns in both models. Therefore, in order to get a more objective view on the performance of the two approaches, we have also presented an equal-weight strategy for portfolio allocation - a strategy that assigns equal weights to all assets in the portfolio and ignores the risk-free bank account. This portfolio of equal weights is a popular benchmark in portfolio optimization literature, and it is generally not trivial to outperform [7].

4.1 Portfolio value movements

First, we examine the movement of the average portfolio values of the two models during the testing period (01.01.2022 to 01.01.2024). The results are shown in Figure 3.

We observe that both the Structured model and the equal weight strategy maintain higher portfolio values than the Unstructured model throughout the whole testing period. We

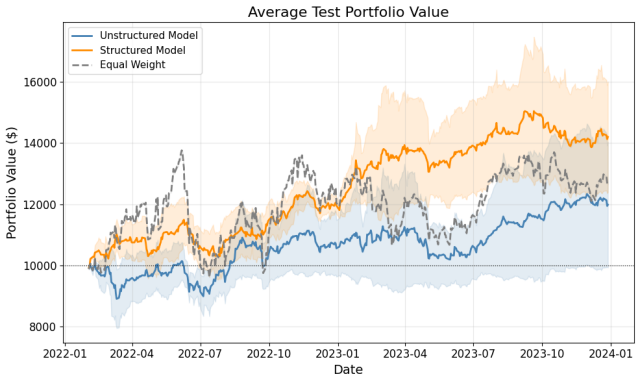

Figure 3: Average portfolio values of the Unstructured and Structured models across the testing period (01.01.2022 to 01.01.2024). Shaded areas display one standard deviation of the portfolio values. The value of a portfolio of equal weights of the assets is also presented.

can also notice that the movement of the portfolio value of the Unstructured model is more closely related to that of the equal-weight strategy. This phenomenon is less present in the Structured model. These findings suggest that the restricted action space of the Structured model, with regard to only trading on the spreads of the pairs, assists it in developing a policy that is more resilient to the overall market movement.

4.2 Performance metrics of the models

We now take a look at more concrete numerical performance metrics of the Structured and Unstructured models. These metrics are presented in Table 3. Standard deviations are displayed in brackets.

Table 3: Performance metrics of the Unstructured and Structured models. The metrics for the equal-weight strategy have also been included. Standard deviation is shown in brackets.

	Unstructured	**Structured**	**EW**
End Value ($)	11,985 (2,052)	14,198 (1,808)	12,670
Ann. Return (%)	9.6 (9.7)	20.1 (8.0)	13.3
Ann. Vol. (%)	20.5 (7.9)	18.0 (5.4)	32.6
Sharpe Ratio	0.104 (0.458)	0.877 (0.324)	0.287
Max Drawdown (%)	−20.2 (1.9)	−13.3 (2.6)	−30.1

We can see that the Structured model displays a considerably better performance out of the three strategies across all 5 metrics. The most notable difference being in the Sharpe ratio, where we have an average of more than 0.773 as compared to the Structured model and an average of more than 0.590 as compared to the equal-weight approach. Another notable advantage is the higher value of maximum drawdown that the Structured model achieves, with 6.9% higher value than the Unstructured model. An important observation is also that the Structured model achieves a lower standard deviation than the Unstructured model across all 5 metrics, except for the maximum drawdown.

The Unstructured model shows an overall poorer performance than its counterpart. The model also fails to outperform the equal-weight baseline in terms of the end value of the portfolio, the annual return, and the Sharpe ratio (although, the high standard deviation of 0.458 suggests the model is capable of outperforming the baseline in terms of Sharpe ratio for some test iterations). However, we can see that the Unstructured model achieves a considerably lower annual volatility by 12.1% and a lower maximum drawdown by 9.9% as compared to the equal-weight approach.

4.3 Weight allocation strategies

Finally, to gain a better insight into the policies of the two models, we take a look at the weight allocation strategies of the two models. Even though we have included a turnover penalty(0.0005) in the reward function of the models, both of them still prefer to make frequent changes to the weight allocated to assets/spreads and cash (an explanation for this could be the overall unstable market movement, which can be observed in Figure 3). Therefore, for better interpretability, we have decided to observe the average weights from the 501 steps in the testing period and 5 testing iterations. We have displayed these weights in Figure 4.

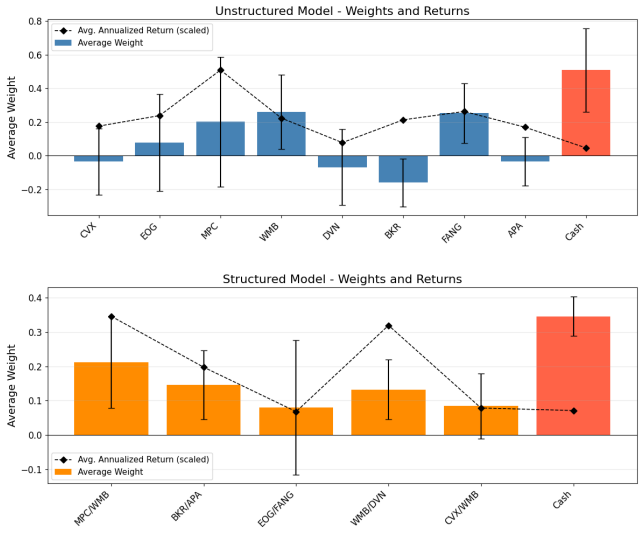

Figure 4: Average weights allocated to assets/spreads and risk-free bank account (cash). Standard deviation of the weights is included. The scaled annual return of the assets/spreads and cash is displayed with a black line.

We see that the Unstructured model has a preference to assign higher weights of around 0.20, 0.27, and 0.20 to the assets MPC, WMP, and FANG, respectively. On the other hand, it has a high tendency to short-sell the assets DVN and BKR by assigning negative weights of -0.08 and -0.17. The model shows a reliance on the risk-free bank account by assigning a high average weight of 0.51. The weights with the highest variance are the ones associated with MPC, EOG, and the risk-free bank account.

The Structured model assigns positive weights to all 5 spreads. The pair with the highest weight of around 0.21 is MPC/WMB. The model tends to assign low weights of around 0.07 and 0.08 to the pairs EOG/FANG and

CVX/WMB. The Structured model also displays a high preference to invest in the cash-free bank account by assigning an average weight of 0.35. The weights with the highest variance are the ones associated with pairs EOG/FANG and MPC/WMB.

Overall, both models have a high tendency to invest in the risk-free bank account. However, the Unstructured model tends to assign a higher weight to it. Additionally, the Unstructured model tends to rely on short selling by assigning negative weights to several assets, while the Structured model avoids short selling(performed if we do not count the implicit short-selling performed within each pair). Both models have high variance in their weight assignment, with the Unstructured model displaying a notably higher variance in the weight associated with the cash-free bank account as compared to the Structured model.

In Figure 4, we have also shown the annual return of the assets, spreads, and the risk-free bank account. The motivation for this unusual choice is the desire to interpret and justify the weight allocation performed by the models. In order to be comparable to the weights, these returns have been scaled with the following formula:

$$\tilde{r}_i = r_i \cdot \frac{\max_j |w_j|}{\max_j |r_j|}, \tag{9}$$

where r_i is the annualized return of asset i, and w_j is the average weight of asset j.

We can see that there is a correlation between the assigned weights and the annual profits of the stocks and spreads. Figure 4 also suggests this correlation to be stronger in the Structured model. However, this relationship does not hold for the risk-free bank account in both models.

4.4 Analysis across Different Number of Asset Pairs

We now make a comparison between the Unstructured and Structured models across a different number of asset pairs, namely 1, 3, 5, and 10. We have decided to focus on the annual returns and Sharpe ratios as performance metrics. The choice for annual returns is motivated by their direct relation to the objective of the two models. Sharpe ratios were chosen because they put returns of the portfolio in the context of its volatility and thus provide a notion of risk.

Since both metrics are heavily reliant on the stocks the models trade on, and these stocks change with the different number of pairs, we have also decided to include the equal-weight benchmark in our analysis.

The annual returns of the models are presented in Figure 5. We can see that the Structured model obtains the highest annual returns of 23.7%, 20.1%, and 23.9% for 3, 5, and 10 pairs, respectively. However, the equal-weight benchmark achieves the highest return for one pair of assets with a value of 27.4%. The Unstructured model achieves the lowest annual returns in all four cases, with values of 7.9%, 6.9%, 9.6%, and 14.2%.

As we increase the number of pairs, the variance of the Unstructured model increases, starting from 4.5% and reaching 15.3%. This trend is not present in the Structured model, where the variance for the case of 5 pairs is lower than the variances in the 3 and 10 pair cases.

A notable result is that the Structured model achieves annual returns at least 6.8 percentage points higher than the equal-weight benchmark in the 3, 5, and 10 pair scenarios. However, these annual returns display high variances, reaching up to 20.8% in the case of 10 pairs.

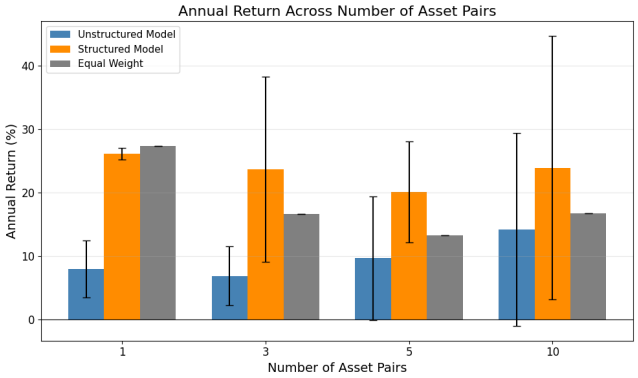

Figure 5: Annual returns of the Unstructured and Structured models across different numbers of asset pairs. Equal-weight strategy values have been included. Standard deviations are also displayed.

We now take a look at the Sharpe ratios across a different number of pairs. The Sharpe ratios are displayed in Figure 6. We observe that the Structured model again achieves the highest values for the 3, 5, and 10 pair cases. These values are 0.726, 0.877, and 0.465, respectively. The best Sharpe ratio for one pair is achieved by the Unstructured model, with a value of 1.583, caused by investing large amounts in the risk-free bank account. However, this value is accompanied by a high variance of 1.712.

With the increasing number of pairs, the variance of the Unstructured model drops from 1.712 in the case of 1 pair to 0.200 in the case of 10 pairs. We again observe that no pattern is present for the variance of the Structured model. Its highest variance is 0.499 for the case of 3 pairs.

It is notable that the Structured model manages to achieve Sharpe ratios of 0.336 and 0.590 more than the equal-weight benchmark in the case of 3 and 5 pairs, respectively.

5 Discussion

We now dive into the discussion of the results and the implications they might have in the world of pairs trading and portfolio optimization. We also take a critical look at the validity of our results.

5.1 Models comparison

The portfolio value movements throughout the testing period and the performance of the two models on the 5 portfolio evaluation metrics suggest the Structured model to be much more suitable for the task of portfolio optimization via pairs trading than the Unstructured model. Going from the Unstructured to the Structured model, we manage to achieve an increase in Sharpe ratio of more than 0.773 and an increase in annual volatility of 9.5%. The Structured model

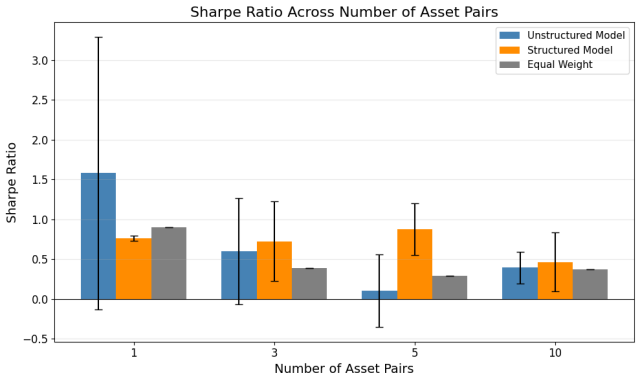

Figure 6: Sharpe ratios of the Unstructured and Structured models across different numbers of asset pairs. Equal-weight strategy values have been included. Standard deviations are also displayed.

also presents an overall lower standard deviation across the 5 metrics, which can be explained by the more stable learning curve of the model. This showcases the advantage of the restriction of the action space to only trading on the spreads of the pairs, and the inclusion of the pairs' statistics in the state of the model.

However, the considerably poorer performance of the Unstructured model as compared to the equal-weight benchmark might deem it inappropriate for comparison and could be an indication of a bias towards the Structured model during the development of the two strategies.

5.2 Applicability of the Structured model

A good way to gain more insight into the applicability of the Structured model is its comparison with the equal-weight benchmark [7]. It is promising that the model manages to outperform the equal-weight strategy by 6.8% in terms of annual return and by 0.590 in terms of Sharpe ratio. A factor that might gain the interest of risk-averse investors is that the annual volatility is lower by 14.6% as compared to the benchmark. Another good indication is the steady increase in portfolio value indicated by the maximum drawdown of only -13.3%.

However, recent research on pairs trading strategies has achieved a mean annual return of 24.8% and remarkable Sharpe ratios of above 2.04 [9]. Although these results were not obtained in the context of portfolio optimization, they might still suggest that the Structured model performs suboptimally. Another factor that hinders the applicability of the model is its high variance, especially regarding the Sharpe ratio, where we observe a standard deviation of 0.324.

One last point is that the current model has drawbacks in terms of transparency - it is not trivial to justify the individual weight allocations the agent performs. As such, this current version of the Structured model is likely unsuitable for deployment in the real world.

5.3 Performance across a different number of asset pairs

Our results showcased that the Structure model outperforms both the Unstructured model and the equal-weight benchmark in terms of annual returns and Sharpe ratios for the case of 3, 5, and 10 pairs. This is a good indication that the model generalises well with a different number of asset pairs. However, the high volatility in both metrics for the cases of 3 and 5 pairs could be an indication of instability of the model. With regards to the case of 1 asset pair, the Structured model falls behind the equal-weight benchmark with 1.2% lower annual return and 0.139 lower Sharpe ratio. The reason for that is the restriction on only trading with one pair and the risk-free bank account.

No notable patterns between performance and the number of pairs were discovered for the Structured model. On the other hand, the Unstructured model displayed an increase in variance in annual returns with the increasing number of pairs. This increase in instability can be explained by the increasing size of the unrestricted action space of the model. The decreasing variance in Sharpe ratios showcases that the Unstructured model manages to achieve a better balance between returns and volatility of the portfolio when we increase the number of available assets.

Since the hyperparameters of the two models were tuned for the case of 5 pairs and reused in the other scenarios, the conclusions in this section should not be taken with absolute certainty. Thus, further investigation with properly optimized models for all four scenarios would prove beneficial.

6 Responsible Research

6.1 Ethical Implications

One of the ethical implications we consider is the impact that the Structured and Unstructured models might have on market stability. For example, deploying the agents at scale could lead to sudden liquidity drains(insufficient volumes of particular stocks) in the market, as the models trade on the limited set of assets that comprise the selected pairs and assume any amount of the stock could be bought or sold. Additionally, the models can exhibit unstable behaviour if the presumed cointegrated pairs change their behaviour and the assets within them decouple.

Another serious issue inherent to reinforcement learning models in general is explainability. Financial supervisors require institutions to transparently justify exactly why a trading decision was made. While the regulation of markets with active AI agents is already a widely recognized problem [1], it is particularly challenging for our models because the complex logic behind how the agents distribute portfolio weights cannot be easily explained.

6.2 Reproducibility, Replicability and Integrity

The code base uses fixed, logged random seeds in order to ensure reproducibility. Additionally, the full code base that includes all experimental setup and results reporting is available at [GitHub/RadoslavGeorgiev71/ Multiple-Pairs-Trading-for-Portfolio-Optimization-with-Reinforcement-Learning].

The taken integrity measures include the clear separation of training and test data: pair selection, spread creation, and training are performed solely on training data; as well as reporting results by looking at statistics over several train-test sessions, instead of relying on a single instance.

One of the main shortcomings of this study is the projected lack of replicability. While all results are reproducible, their high variance suggests that slight changes in code and data can lead to meaningful differences.

6.3 Data and Privacy

This study relies exclusively on historical, publicly available data retrieved from Yahoo Finance for S&P 500 stocks. Because the dataset contains no personally identifiable information or proprietary user data, this research poses no privacy or ethical risks regarding data collection.

6.4 LLM Use Disclosure

We used a large language model as a writing assistant for the draft of the paper. This use includes the correction of grammar and spelling mistakes, as well as the rephrasing of existing text. In more technical sections like methodology, a model was occasionally used to generate the section structure based on the given code.

The assistance of LLMs was also used for the development of the code base. This was mainly restricted to experimentation with different ideas by augmenting already existing code, and assistance in generating appropriate tables and figures. However, all code has been manually inspected and debugged.

7 Conclusions and Future Work

This paper developed two PPO Reinforcement Learning models for the purposes of portfolio optimization with pairs trading. The first model, called Unstructured, is completely unaware of the pair relationships between the assets and can allocate weights between them without restrictions. On the other hand, the second model, called the Structured, is restricted to only trading on the spreads of the assets within the pairs, but has additional information in terms of statistics for the pairs. We showed that the added pairs trading structure to the second model helps it achieve considerably better training stability and performance across all 5 portfolio evaluation metrics. The results of the Structured model were also accompanied by lower variance in the 5 metrics. The Structured model also managed to consistently outperform an equal-weight baseline portfolio. Additionally, the strategy displayed good performance in terms of annual returns and Sharpe ratios across a different number of asset pairs, but no notable relations between performance and the number of pairs were discovered. The results showcase that a multi-pair RL model approach to portfolio optimization might be promising and emphasize the need for further investigation.

One future prospect for the study is fine-tuning the hyperparameters of the two models for different numbers of asset pairs. This will give more credibility to the study in that dimension and possibly manage to discover some relations between performance and number of asset pairs. Another direction to examine is to make a comparison in performance when we use pairs from one market sector, and when we make use of pairs across different sectors. This could lead to a good evaluation of how well the model utilizes relations between the traded pairs.

References

[1] Alessio Azzutti and A Azzutti AlessioAzzutti. Ai governance after mifid ii: beyond (mere) technological neutrality? *ERA Forum 2026 27:1*, 27:7–31, 2 2026.

[2] Faik Bilgili. Munich personal repec archive stationarity and cointegration tests: Comparison of engle-granger and johansen methodologies stationarity and cointegration tests: Comparison of engle-granger and johansen methodologies. 1998.

[3] David A. Dickey and Wayne A. Fuller. Distribution of the estimators for autoregressive time series with a unit root. *Journal of the American Statistical Association*, 74:427–431, 1979.

[4] Robert F. Engle and Clive W.J. Granger. Co-integration and error correction: Representation, estimation, and testing. *Applied Econometrics*, 39:107–135, 2015.

[5] Saeid Fallahpour, Hasan Hakimian, Khalil Taheri, and Ehsan Ramezanifar. Pairs trading strategy optimization using the reinforcement learning method: a cointegration approach. *Soft Computing 2016 20:12*, 20:5051–5066, 8 2016.

[6] Evan Gatev, William N. Goetzmann, and K. Geert Rouwenhorst. Pairs trading: Performance of a relative-value arbitrage rule. *The Review of Financial Studies*, 19:797–827, 10 2006.

[7] Matteo Gelmini and Pierpaolo Uberti. The equally weighted portfolio still remains a challenging benchmark. *International Economics*, 179:100525, 10 2024.

[8] Abhishek Gunjan and Siddhartha Bhattacharyya. A brief review of portfolio optimization techniques. *Artificial Intelligence Review 2022 56:5*, 56:3847–3886, 9 2022.

[9] Chulwoo Han, Zhaodong He, and Alenson Jun Wei Toh. Pairs trading via unsupervised learning. *European Journal of Operational Research*, 307:929–947, 6 2023.

[10] Weiguang Han, Boyi Zhang, Qianqian Xie, Min Peng, Yanzhao Lai, and Jimin Huang. Select and trade: Towards unified pair trading with hierarchical reinforcement learning. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1:4123–4134, 8 2023.

[11] Le Trung Hieu. Deep reinforcement learning for stock portfolio optimization. *International Journal of Modeling and Optimization*, 10:139–144, 12 2020.

[12] Nicolas Huck and Komivi Afawubo. Pairs trading and selection methods: is cointegration superior? *Applied Economics*, 47:599–613, 2 2015.

[13] Søren Johansen. Statistical analysis of cointegration vectors. *Journal of Economic Dynamics and Control*, 12:231–254, 6 1988.

[14] Jun Kevin and Pujianto Yugopuspito. Hybrid lstm and ppo networks for dynamic portfolio optimization. *Proceedings of the 2025 8th Artificial Intelligence and Cloud Computing Conference*, 1:309–319, 12 2025.

[15] Kiseop Lee, Tim Leung, and Boming Ning. A diversification framework for multiple pairs trading strategies. *Risks 2023, Vol. 11, Page 93*, 11:93, 5 2023.

[16] Timothy P. Lillicrap, Jonathan J. Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. *4th International Conference on Learning Representations, ICLR 2016 - Conference Track Proceedings*, 9 2015.

[17] Yan-Xia Lin, Michael Mccrae, and Chandra Gulati. Loss protection in pairs trading through minimum profit bounds: A cointegration approach. 2006.

[18] Robert C. Merton. Lifetime portfolio selection under uncertainty: The continuous-time case. *The Review of Economics and Statistics*, 51:247, 8 1969.

[19] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. *Nature 2015 518:7540*, 518:529–533, 2 2015.

[20] Alessia Naccarato, Andrea Pierini, and Giovanna Ferraro. Markowitz portfolio optimization through pairs trading cointegrated strategy in long-term investment. *Annals of Operations Research 2019 299:1*, 299:81–99, 4 2019.

[21] Boming Ning and Kiseop Lee. Advanced statistical arbitrage with reinforcement learning. *International Journal of Financial Engineering*, 12, 3 2024.

[22] Paul A. Samuelson. Lifetime portfolio selection by dynamic stochastic programming. *The Review of Economics and Statistics*, 51(3):239–246, 1969.

[23] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov Openai. Proximal policy optimization algorithms. 7 2017.

[24] William F. Sharpe. Mutual fund performance. *The Journal of Business*, 39(1):119–138, 1966.

[25] David Silver, Guy Lever, Nicolas Heess, Thomas Degris, Daan Wierstra, and Martin Riedmiller. Deterministic policy gradient algorithms. In Eric P. Xing and Tony Jebara, editors, *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pages 387–395, Bejing, China, 22–24 Jun 2014. PMLR.

[26] G. E. Uhlenbeck and L. S. Ornstein. On the theory of the brownian motion. *Physical Review*, 36:823, 9 1930.

[27] Cheng Wang, Patrik Sandas, and Peter Beling. Improving pairs trading strategies via reinforcement learning. *2021 International Conference on Applied Artificial Intelligence, ICAPAI 2021*, 5 2021.

[28] Hongyang Yang, Xiao Yang Liu, Shan Zhong, and Anwar Walid. Deep reinforcement learning for automated stock trading: An ensemble strategy. *ICAIF 2020 - 1st ACM International Conference on AI in Finance*, 20, 10 2020.