

Article

Reinforcement Learning Pair Trading: A Dynamic Scaling Approach

Hongshen Yang * and Avinash Malik

Department of ECSE, The University of Auckland, Auckland 1010 , New Zealand; avinash.malik@auckland.ac.nz
* Correspondence: hyan212@aucklanduni.ac.nz

Abstract: Cryptocurrency is a cryptography-based digital asset with extremely volatile prices. Around USD 70 billion worth of cryptocurrency is traded daily on exchanges. Trading cryptocurrency is difficult due to the inherent volatility of the crypto market. This study investigates whether Reinforcement Learning (RL) can enhance decision-making in cryptocurrency algorithmic trading compared to traditional methods. In order to address this question, we combined reinforcement learning with a statistical arbitrage trading technique, pair trading, which exploits the price difference between statistically correlated assets. We constructed RL environments and trained RL agents to determine when and how to trade pairs of cryptocurrencies. We developed new reward shaping and observation/action spaces for reinforcement learning. We performed experiments with the developed reinforcement learner on pairs of BTC-GBP and BTC-EUR data separated by 1 min intervals ($n = 263,520$). The traditional non-RL pair trading technique achieved an annualized profit of 8.33%, while the proposed RL-based pair trading technique achieved annualized profits from 9.94% to 31.53%, depending upon the RL learner. Our results show that RL can significantly outperform manual and traditional pair trading techniques when applied to volatile markets such as cryptocurrencies.

Keywords: pair trading; reinforcement learning; algorithmic trading; deep learning; cryptocurrency



Citation: Yang, Hongshen, and Avinash Malik. 2024. Reinforcement Learning Pair Trading: A Dynamic Scaling Approach. *Journal of Risk and Financial Management* 17: 555.
<https://doi.org/10.3390/jrfm17120555>

Academic Editor: Xianrong (Shawn) Zheng

Received: 8 November 2024

Revised: 28 November 2024

Accepted: 6 December 2024

Published: 11 December 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Arbitrage is a subdomain of financial trading that profits from price discrepancies in different markets (Dybvig and Ross 1989). Pair trading is one of the well-known arbitrage trading methods in financial markets. Arbitrageurs identify two highly correlated assets to form a pair. When a price discrepancy happens, they buy the underpriced asset and sell the overpriced correlated asset to profit from the mean reversion of the prices. With the rise of high-frequency trading, the ability to conduct fast and accurate analyses has become critical. Arbitrage requires practitioners to constantly analyze the market conditions at the fastest speed possible, as arbitrageurs must compete for transitory opportunities (Brogaard et al. 2014). Therefore, we explore how Artificial Intelligence (AI) can enhance the process of pair trading, focusing on the speed and adaptability of decision-making.

Reinforcement Learning (RL) is a captivating domain of AI. The idea of RL is to let the agent(s) learn to interact with an environment. The agent should learn from the environment's responses to optimize its behavior (Sutton and Barto 2018). If we view the financial market from the perspective of the RL environment, actions in the financial market are investment decisions. By allowing agents to adapt dynamically to market conditions, RL has the potential to overcome the limitations of static, rule-based strategies in volatile and complex financial environments. For gaining profits, arbitrageurs are incentivized to train agents to produce lucrative investment decisions, and RL facilitates agents' learning process from the profit/loss of the market.

The combination of RL and various financial trading techniques is still evolving rapidly. There has been some work in RL infrastructural construction (Liu et al. 2021, 2022a, 2022b)

and some experiments in profitable RL agent training ([Meng and Khushi 2019](#); [Pricope 2021](#); [Zhang et al. 2020](#)). Trading actions in traditional pair trading follow static rules. In reality, the complexity of financial markets should allow more flexibility in the decision-making process. An experienced trader might analyze market conditions to make informed decisions. However, it is not feasible to output efficient decisions at short, intermittent intervals 24/7. RL algorithms enable a fast-track decision-making process for analyzing trading signals and generating trading actions.

Designing a high-frequency trading system based on RL requires addressing critical challenges. The first challenge is how to construct an RL environment that accommodates RL agents for arbitrage. The second challenge involves identifying compatible instruments with historical correlations to form profitable pairs. The third challenge concerns timing. Instead of blindly following preset rules, the system requires flexibility in choosing investment timings to achieve greater profitability. The final challenge involves investment quantity. Since investment opportunities vary in quality, a critical consideration is whether RL agents can replicate decision-making capabilities comparable to the scrutiny applied by experienced traders.

This paper investigates key questions centered around the application of Reinforcement Learning (RL) in pair trading. To address the fast decision-making requirements in a high-frequency trading environment, we constructed a tailored RL environment for pair trading and fine-tuned reward shaping to encourage the agent to make profitable decisions. The contributions of this work are as follows: (1) the construction of an RL environment specifically designed for quantity-varying pair trading; (2) the proposal of a novel pair trading method that incorporates adaptive investment quantities to capture opportunities in highly volatile markets; (3) the use of a grid search technique to fine-tune hyperparameters for enhanced profitability; (4) the introduction of an RL component for market analysis and decision-making in pair trading, along with a novel RL model optimized for investment quantity decisions.

The structure of the paper is arranged as follows: the background and related work are introduced in Sections 2 and 3. The methodology is presented in Section 4. Experiments and results are included in Section 5. A discussion of the results and conclusions is provided in Section 6.

2. Background

First, we define the basic terms of financial trading. A *long* position is created when an investor uses cash to buy an asset, and a *short* position is created when an investor sells a borrowed asset. The portfolio is the investor's total holding, including long/short position and cash. *Transaction cost* is a percentage fee payable to the broker for any long/short actions. Finally, *risk* is defined as the volatility of the portfolio.

2.1. Traditional Pair Trading

Classical pair trading consists of two distinct components known as *legs*. A *leg* represents one side of a trade in a multi-contract trading strategy. Under the definition of pair trading, "longing the first asset and shorting the second asset" is called a *long leg*, and "shorting the first asset and longing the second asset" is called a *short leg*. The two assets are always bought and sold in opposite directions in pair trading. Therefore, the overall pair trading strategy is considered to be *market neutral* because the profits from the *long* position and the *short* are offset by the direction of the overall market. [Gatev et al.'s \(2006\)](#) work is the most cited traditional pair trading method. It follows the OODA (observe, orient, decide, and act) loop ([Fadok et al. 1995](#)). Before entering the market, the first step is to choose the proper assets in a pair. The Sum of Squared Deviation (SSD) is the measurement calculated from the prices of assets i and j . Through exhaustive searching in a formation period T , the assets with the smallest SSD are bound as a pair Equation (1).

$$SSD_{p_i, p_j} = \sum_{t=1}^T (p_i - p_j)^2. \quad (1)$$

- **Observe** is the process of market analysis. The price of assets in pairs is collected and processed. The price difference ($p_i - p_j$) is called spread S . The arbitrageurs *observe* the current positions and spread of the current market.
- **Orient** is the process exploring what could be done. Three possible actions for pair trading are long leg, short leg, and close position, as defined above.
- **Decide** what action to take. Position opening triggers when the price difference deviates too much. This is indicated by the spread movement beyond an open threshold. Position closing happens when the spread reverts back to some closing threshold. [Gatev et al. \(2006\)](#) adopted two times the standard deviation of the spread as the opening threshold and the price crossing as the closing threshold. In practice, the threshold varies according to the characteristics of the financial instrument.
- **Act** once the decision is made. The long leg orders us to buy asset i and sell asset j . The short leg orders us to sell asset i and buy asset j . Closing a position means clearing all the active positions to hold cash only.

A graphical visualization of pair trading is presented in Figure 1. Figure 1a shows the market interactions according to the Spread (S) and thresholds. A position is opened whenever the spread deviates beyond the open threshold. The position closure happens when the spread reverts below the close threshold. Figure 1b, which shares the same time axis with (a), is a stretched view of (a). It presents the corresponding actions with the crossing of Spread (S) and zones. The spread deviations are classified into zones based on the Spread (S), Open-Threshold (OT), and Close-Threshold (CT) Equation (2):

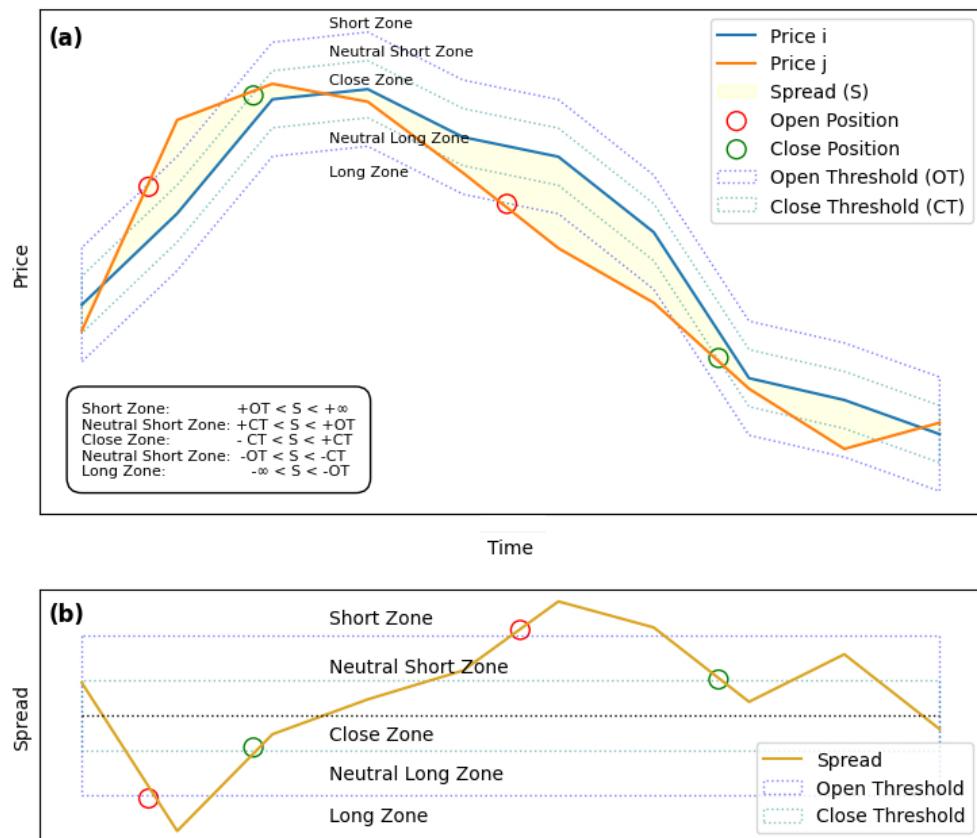


Figure 1. Stretched pair trading view of price distance between p_i and p_j . Figure (b), which shares the same time axis with (a), is a stretched view of (a). It presents the corresponding same actions with the crossing of Spread (S) and zones in two different views.

$$\left\{
 \begin{array}{ll}
 \text{Short Zone:} & +OT < S < +\infty, \\
 & \text{spread deviates beyond open threshold} \\
 \text{Neutral Short Zone:} & +CT < S < +OT, \\
 & \text{spread deviates between open and close threshold} \\
 \text{Close Zone:} & -CT < S < +CT, \\
 & \text{spread reverts between close thresholds} \\
 \text{Neutral Long Zone:} & -OT < S < -CT, \\
 & \text{spread deviates between open and close threshold} \\
 \text{Long Zone:} & -\infty < S < -OT, \\
 & \text{spread deviates below open threshold}
 \end{array}
 \right. . \quad (2)$$

2.2. Reinforcement Learning

Reinforcement Learning (RL) is used to train an agent to maximize rewards while interacting with an environment (Sutton and Barto 2018). The environment for RL is required to be a Markov Decision Process (MDP) (Bellman 1957), which means it is modeled as a decision-making process with the following elements (State (S), Action (A), Transition (P_A), Reward (R_A)). The goal is to train the agent to develop a policy (π) that fulfills an objective, e.g., maximizing profits in a trade. At every trading interval t , according to state S , that the agent observes, action A is chosen based on policy π . The environment rewards/punishes the state transition of $S_t \rightarrow S_{t+1}$ with environment reward r . If we assume γ to be the discount factor for the time-value discount of future reward, RL trains a policy π that maximizes the total discounted reward G_t , as shown in Equation (3):

$$G_t = \sum_{i=0}^{\infty} \gamma^i r_{t+i}. \quad (3)$$

RL algorithms can broadly be classified based on three criteria: value/policy-based, on/off policy, and actor/critic network (AlMahamid and Grolinger 2021). Value-based methods estimate state-action value functions for decision-making. Policy-based methods directly learn action selection policies. The on-policy method requires data generated by the current policy, and the off-policy method is capable of leveraging past experiences from potentially different policies. Moreover, actor–critic architectures, where an actor–network proposes actions and a critic network evaluates, have shown a better performance in facilitating policy improvement through this feedback loop. Most recent research favors an actor–critic architecture instead of actor-only or critic-only methods for better performance (Meng and Khushi 2019; Zhang et al. 2020). Therefore, only actor–critic algorithms are adopted in this study.

Based on the RL classification criteria, some representative algorithms have been selected for this study including Deep Q-Learning (DQN) (Mnih et al. 2013), Soft Actor Critic (SAC) (Haarnoja et al. 2019), Advantage Actor-Critic (A2C) (Sutton and Barto 2018), Proximal Policy Optimization (PPO) (Schulman et al. 2017). Diversified RL algorithms are experimented with to choose the most effective one in pair trading.

3. Related Work

3.1. Reinforcement Learning in Algorithmic Trading

Reinforcement learning in AlphaGo captured the world's attention in 2016 by participating in a series of machine versus human competitions on the board game GO (Silver and Hassabis 2016). Surprisingly, the research regarding RL in the financial market started long before that. Recurrent reinforcement learning studies were the mainstream works (Bertoluzzo and Corazza 2007; Gold 2003; Maringer and Ramtohul 2012; Zhang and Maringer 2016) in the early stage of financial trading. After the upsurge of AlphaGo, some significant advancements were brought to RL trading as well; Huang (2018) re-described the Markov Decision Process (MDP) financial market as a game process to incorporate RL as

a financial trading game (Huang 2018). Pricope (2021) proposed deep RL agents to develop profitable high-frequency trading strategies with sequential model-based optimization tuning the hyperparameters. With the recent development, newer RL models such as Deep Q-Learning (DQN), Policy Gradients (PG), and Advantage Actor-Critic (A2C) have also been introduced by researchers (Meng and Khushi 2019; Mohammadshafie et al. 2024; Zhang et al. 2020) for financial trading. A noteworthy research work is that of the FinRL group in the infrastructures and ensemble learning mechanism (Liu et al. 2021, 2022a, 2022b).

3.2. Reinforcement Learning in Pair Trading

Reinforcement Learning, in combination with pair trading, is not an untapped domain. RL has ameliorated multifaceted aspects of the traditional method of pair trading brought up by Gatev et al. (2006). The RL technique Ordering Points To Identify The Clustering Structure contributed to the pair selection stage by leveraging a clustering algorithm to produce better pair choices (Sarmento and Horta 2020). Vergara and Kristjanpoller (2024) traded deep reinforcement learning in the cryptocurrency market with ensemble practice, combining classical pair trading with the RL framework. reCurrent Reinforcement LEarning methoD for paIrs Trading (CREDIT), an algorithm that takes into consideration both profitability and risks, was engineered by Han et al. (2023). Reward shaping is also an interesting area where some work has been done in RL trading (Lucarelli and Borrotti 2019; Wang et al. 2021). Kim and Kim's work in (2019) is the most recent RL pair trading method. Their focus is on utilizing RL to find the most trading opportunities. Instead of fixed thresholds, the RL agent in Kim and Kim's work produces thresholds for the upcoming trading period. Open, close, and stop-loss thresholds determine the profits of pair trading.

Our work introduces a novel method to combine RL with pair trading. The work of Gatev et al. (2006) is not efficient enough for a high-frequency market. The state-of-the-art method of Kim and Kim (2019) has some deficiencies: (i) it requires the market's volatility to be relatively stable. The RL agent may produce unsuitable thresholds if the market experiences increased volatility. (ii) It lacks flexibility in the investment amount. Opportunities with different qualities are programmed to be invested with the same amount of capital. Once the RL agent determines a threshold, the trading algorithm executes a trade at pre-determined thresholds. We leverage RL to make investment timing and quantity decisions. The adjustable investment amount is a novel feature of our RL pair trading. An RL agent measures how well the investment opportunities are based on observations and invests a larger amount in more promising market conditions. Having another dimension on the investment side should further enhance profitability and reduce risks.

4. Methodology

In this section, we introduce the architecture of the methodology (Figure 2). The architecture includes five steps: (1) *pair formation* for selecting assets to form a tradeable pair (Section 4.1); (2) *spread calculation* utilizing the moving-window technique to extract the spread in a limited retrospective time frame (Section 4.2); (3) *parameter selection* from an historical dataset to decide the most suitable hyperparameters for pair trading (Section 4.2); (4) *RL trading* by allowing RL to decide the trading timing and quantity in pair trading (Section 4.4); (5) *investment action* for taking the actions produced from RL trading into market execution.

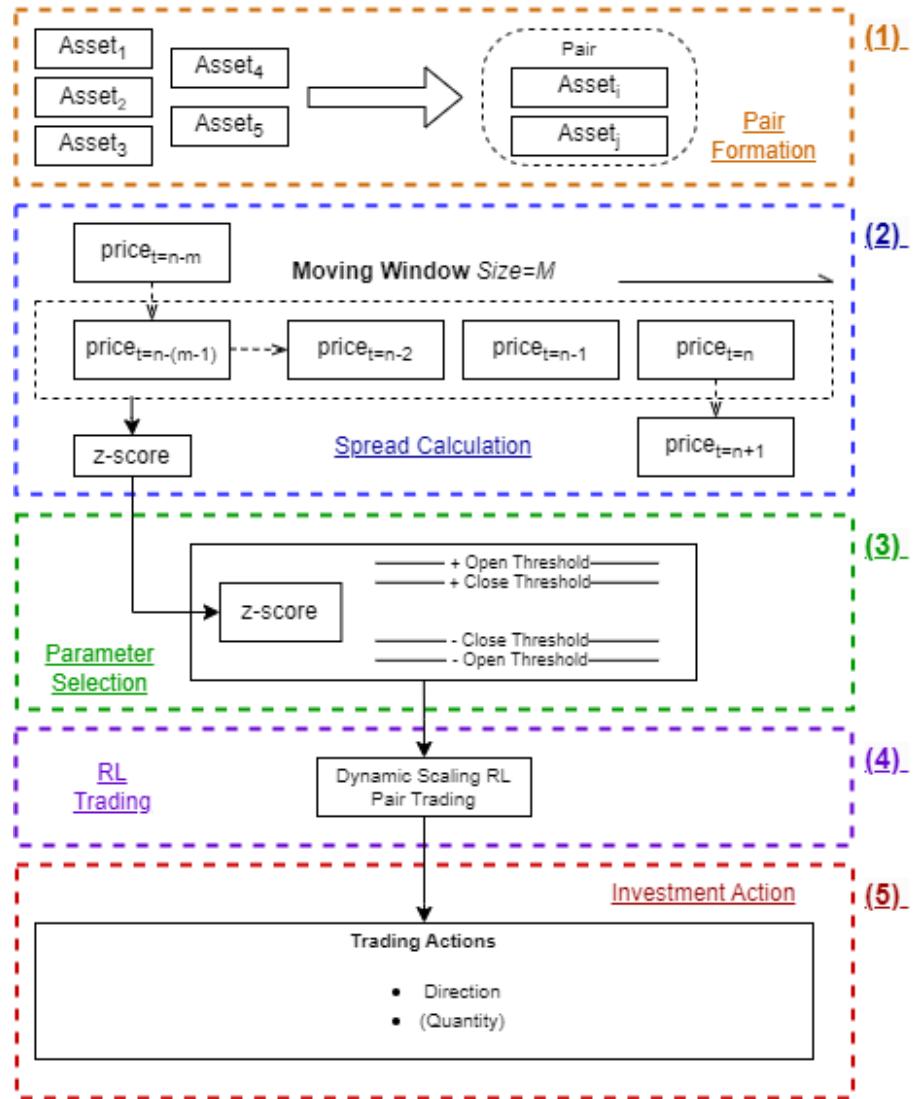


Figure 2. Architecture of trading strategies.

4.1. Pair Formation

Pairs are selected based on two criteria: correlation and cointegration. The widely adopted Pearson's correlation ([Do and Faff 2010; Perlin 2007](#)) is given by

$$\rho_{X,Y} = \frac{\text{cov}(X, Y)}{\sigma_X \sigma_Y}, \quad (4)$$

where $\rho_{X,Y}$ is the correlation coefficient between assets X and Y , $\text{cov}(X, Y)$ is the covariance of X and Y , and σ_X and σ_Y are the standard deviations of X and Y , respectively. The Engle–Granger cointegration test ([Burgess 2003; Dunis and Ho 2005](#)) involves two steps. First, the linear regression is performed:

$$Y_t = \alpha + \beta X_t + \epsilon_t, \quad (5)$$

where Y_t and X_t are the asset price series, α and β are the regression coefficients, and ϵ_t is the residual term. The second step tests the residuals ϵ_t for stationarity using an Augmented Dickey–Fuller (ADF) ([Dickey and Fuller 1979](#)) test. The ADF test regression is given in Equation (6):

$$\Delta \epsilon_t = \gamma \epsilon_{t-1} + \sum_{i=1}^p \delta_i \Delta \epsilon_{t-i} + \nu_t, \quad (6)$$

where $\Delta\epsilon_t$ is the first difference of the residuals, γ is the coefficient to be tested for stationarity, p is the number of lagged difference terms included, and ν_t is the error term. If γ is significantly different from zero, the residuals are stationary, indicating co-integration.

A moving window is applied to historical pricing data, as shown in Figure 3. In this figure, the blue line represents the historical prices, while the dashed boxes illustrate the moving window. During the selection phase, averaged correlation and co-integration batches are employed to ensure that the selected assets exhibit a strong, long-term statistical relationship.

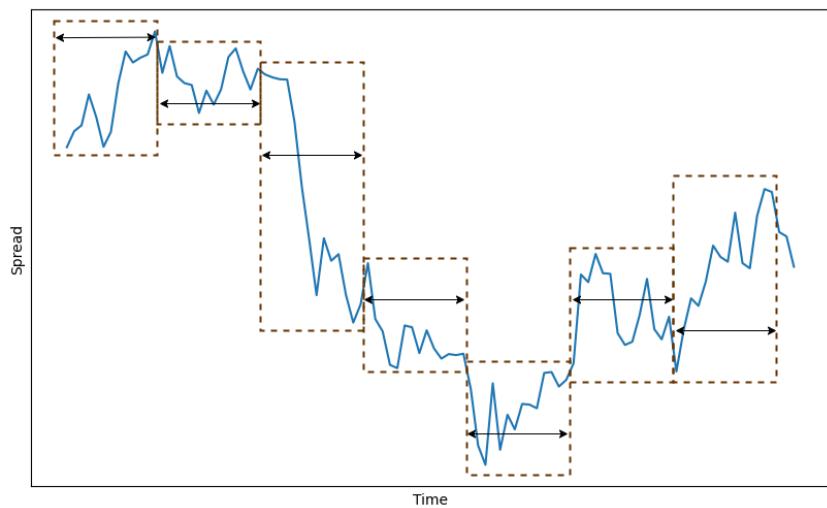


Figure 3. Window-size cut for correlation and co-integration testing.

4.2. Spread Calculation

The second step is a moving window mechanism to capture the spread movement (Figure 2 Step-2). Spread ϵ_t is calculated at every selected trading interval (e.g., every five minutes) and is the error term s from a regression between the two prices p_i and p_j . β_0 and β_1 are the intercept and slope, respectively, which follow a normal distribution with a mean of 0 and a standard deviation of σ Equation (7):

$$p_i = \beta_0 + \beta_1 \times p_j + s_i \sim N(0, \sigma^2). \quad (7)$$

We normalize the spread with z-score Equation (8) to scale the spread into constant mean and standard deviation. The mean of the spread in the sliding window is represented as \bar{s} :

$$Z = \frac{s - \bar{s}}{\sigma_s}. \quad (8)$$

4.3. Parameter Selection

Three parameters to be explored are window size, open threshold, and close threshold. *Window size* $\in \mathbb{Z}^+$ is the number of historical samples in the moving window. *Thresholds* $\in \mathbb{Q}^+$ are the entry and exit signals of trading actions that are highly linked to market conditions.

Excessively wide thresholds suit more volatile markets, and conservatively narrow thresholds result in smaller but steadier wins. The combination of parameters of the highest profitability $\langle \text{Window Size}, \text{Open Threshold}, \text{Close Threshold} \rangle$ are selected from a search pool through a grid search in practice. Gatev et al. (2006) adopted 2 times the standard deviation as the open threshold and the deviation crossing point as the close threshold (Figure 1). However, on the one hand, the parameters should vary with the arbitrage instruments as well as the market condition. Hence, the window-sliding mechanism is incorporated to reflect the heterogeneity of the pricing variance (Mandelbrot 1967).

4.4. Reinforcement Learning Pair Trading

After we run the grid search of window-sliding pair trading, the next problem concerns “when” and “how much” to trade. Pair trading results from following pre-set rules (à la [Gatev et al. 2006](#)), which are obtainable using window-sliding pair trading. However, we want to know if RL produces better investment decisions than blindly following the rules. Therefore, the most profitable parameter combination is passed onto further RL-based pair trading so that we can compare the results between RL-based pair trading and non-RL pair trading.

4.4.1. Observation Space

Observation space stands for the information an RL agent observes. The agent observes market information to make decisions. The observations adopted for our RL environment are as follows: $\langle \text{Position}, \text{Spread}, \text{Zone} \rangle$.

- **Position** $\in [-1, 1]$: Position stands for the current portfolio value. Position is a percentage measuring the direction of investment (c.f. Figure 4). As in pair trading, we define longing the first asset with shorting the second asset as ‘holding a long leg’ and the other way around as ‘holding a short leg’. Assuming that we do not use leverage, holding a long leg with a 70% portfolio value gives Position = 0.7. Holding a short leg with a 30% portfolio value gives Position = -0.3. Position 0 means we only hold cash.
- **Spread** $\in \mathbb{R}$: This represents how much the current spread has deviated from the mean (Section 4.2).
- **Zone** $\in \{\text{Zones}\}$: Zone is an important indicator that comes from the comparison between the z-score with the thresholds for signals (Figure 1b). Traditional pair trading ([Gatev et al. 2006](#); [Yang and Malik 2024](#)) takes the zone as the direct trading signal. However, in RL-based pair trading, the zone is an observation for the RL agent to make better decisions.

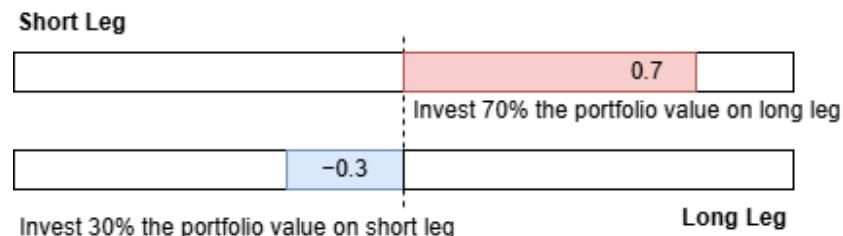


Figure 4. The value of position observation based on investment.

4.4.2. Action Space

Since the pair trading technique is a relatively low-risk strategy, most applications invest with a fixed amount or the complete portfolio value ([Burgess 2003](#); [Huck 2010](#); [Perlin 2009](#)). Meanwhile, it is natural for an experienced trader to invest different amounts based on the quality of opportunities. Opportunities with a higher probability of success are worth more investment. Therefore, we investigate granting the RL agent not only the decision of when to invest but also the freedom to choose the investment amount.

Take $A \in [-1, 1]$ as the action. Similar to the observation space (c.f. Figure 4), the action ranges from -1 to 1, representing the investment as a percentage of the portfolio value in the long leg and short leg directions. Investing 50% of the portfolio as a long leg means $A = 0.5$. Investing 20% of the portfolio in the short leg means $A = -0.2$.

In practice, we have to consider the relationship between the existing position and the next action. We classify the execution of action as *open position*, *adjust position*, or *close position*.

- **Open position** is the action of opening a new position.
- **Close position** is the closure of a position.

- **Adjust position** happens when a previous position is open, and the RL agent wants to open another position. For example, if the current position is a 70% long-leg and the new action is $A = 0.8$, only the extra 10% shall be actioned.

4.4.3. Reward Shaping

The RL reward consists of three components: *action reward*, *portfolio reward*, and *transaction punishment*.

- **Portfolio reward** is the profit/loss from closing a position. The portfolio value V_p only updates when the position closes. V'_p is the position value at the start of a trading period p . Then, upon closing the trade at the end of trading period p , the reward is calculated, as shown in Equation (9).

$$\text{Profit Reward} = V_p - V'_p. \quad (9)$$

- **Action reward** means the agent needs to be rewarded for taking a desired action in the corresponding zone. In general, the agent is free to decide on any action. However, we use *action reward* to encourage the agent to choose desired actions. It rewards the agent for making a desired action in certain zones (Table 1) with some freedom in neutral zones. The stronger the action reward, the more it resembles traditional pair trading.
- **Transaction punishment** is a negative reward for encouraging small adjustments instead of large changes in the position. The punishment is the difference between the action and position. If the current position in observation is P and the action is A , the transaction punishment is Equation (10):

$$\text{Transaction Punishment} = P - A. \quad (10)$$

Table 1. Rewarding behaviors in zones.

Zones	Rewarding Behavior
Short Zone	Short leg
Neutral Short Zone	Short leg or Close
Close Zone	Close
Neutral Long Zone	Long leg or Close
Long Zone	Long leg

4.5. Dynamic Agents

Trading agents in RL environments operate by taking actions within a defined action space based on observations from the state space. This section details the design of two dynamic agent settings, RL_1 and RL_2 , each tailored to address specific aspects of pair trading. These agents leverage the Markov Decision Process (MDP) framework to model the complexities of financial markets, enabling adaptive decision-making in volatile conditions.

In the first setting, RL_1 is tasked with optimizing trade timing and directionality. Pair trading is modeled as a Markov Decision Process (MDP), represented as $(\mathcal{S}_1, \mathcal{A}_1, \mathcal{T}_1, r_1, \gamma_1)$:

- \mathcal{S}_1 represents the state space, including normalized price spreads, historical z-scores, and zone indicator.
- \mathcal{A}_1 defines a discrete action space consisting of three possible actions: initiating a long–short position, closing existing positions, or initiating a short–long position. This allows the agent to determine the optimal direction and timing for trades.
- $r_1(s, a)$ is the reward function, defined as follows:

$$r_1(s_t, a_t) = \begin{cases} \Delta P_t - c, & \text{if a trade is executed;} \\ 0, & \text{if no trade is executed,} \end{cases}$$

where ΔP_t represents the profit or loss from the trade, and c is the transaction cost. The reward function penalizes the agent for transaction costs while directly linking rewards to trade profitability.

- The goal is to maximize the cumulative discounted reward:

$$R_1 = \mathbb{E} \left[\sum_{t=0}^{\infty} \gamma_1^t r_1(s_t, a_t) \right].$$

RL_2 extends RL_1 by shifting focus from trade timing to determining the investment quantity for a given trade opportunity. It models pair trading as an MDP defined by $(\mathcal{S}_2, \mathcal{A}_2, \mathcal{T}_2, r_2, \gamma_2)$:

- \mathcal{S}_2 is the state space, which is identical to RL_1 .
- $\mathcal{A}_2 = [-1, 1]$, where the continuous value represents the investment quantity. Here, 0 stands for no involvement, positive values represent buying, and negative values represent selling.
- $r_2(s, a)$ is the reward function, defined as follows:

$$r_2(s_t, a_t) = \Delta P_t \cdot a_t - c(|a_t|),$$

where $c(|a_t|)$ represents transaction costs proportional to the absolute investment size $|a_t|$. This reward structure incentivizes the agent to optimize both the direction and magnitude of its investment.

- The objective is to maximize the cumulative discounted reward:

$$R_2 = \mathbb{E} \left[\sum_{t=0}^{\infty} \gamma_2^t r_2(s_t, a_t) \right].$$

The primary differences between RL_1 and RL_2 lie in their action spaces and reward functions. RL_1 operates with a discrete action space and focuses on optimizing directional timing and trade management. In contrast, RL_2 uses a continuous action space $\mathcal{A}_2 = [-1, 1]$, enabling it to adjust investment sizes dynamically. The environments are designed to guide the agents by rewarding profitable actions and penalizing costly ones, encouraging effective decision-making for timing and quantity. The exact mechanisms driving these decisions are embedded within the neural network, shaped by the agent's interactions with the environment.

5. Benchmark Results

Next, we carry out experiments using the proposed methodology. We adopt the same dataset and the same parameters for non-RL pair trading and RL pair trading for comparison purposes.

5.1. Experimental Setup

We experiment with window-sliding pair trading and RL pair trading in the cryptocurrency market. The cryptocurrency market is famous for its volatility, easy access, and 24/7 operating time. Data preprocessing follows two steps. Firstly, the evaluation of the correlations among trading pairs for selection of arbitrage candidates; next, searching through the combination of thresholds and retrospective periods for suitable signals.

5.1.1. Datasets

The application of our trading methodology is on Binance, the largest cryptocurrency market.¹ For the best market liquidity, we picked Bitcoin–Fiat currencies under different trading intervals for pair trading. Pair formation criteria are based on Pearson's correlation and augment the Engle–Granger two-step cointegration test (Section 4.1) for quote currencies that follow a similar trend against the base currency (Figure 5). The formation period

is from October 2023 to November 2023, and the test is in December 2023, with trading intervals of 1 min (121,500 entries), 3 min (40,500 entries), and 5 min (24,300 entries), respectively. We exhaustively compared correlation and co-integration for the best pair (Table 2).² Although Binance has quite a few fiat currencies, only the US Dollar (USD), Great British Pound (GBP), Euro (EUR), and Russian Ruble (RUB) display relatively strong liquidity. The pair with the strongest correlation and co-integration is BTCEUR and BTCGBP under a 1 min trading interval (Table 2).

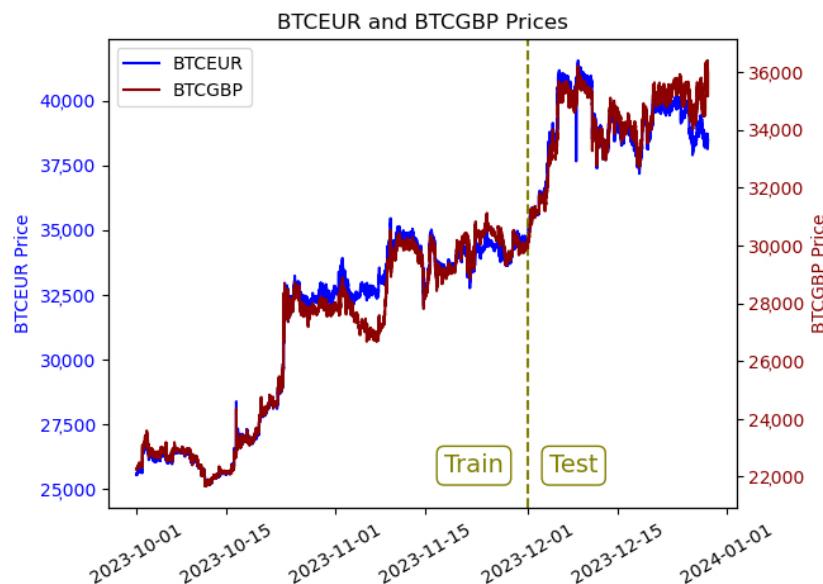


Figure 5. Prices of BTCEUR and BTCGBP.

Table 2. Correlation and co-integration of pair formation.

Pairs	1m coint	1m corr	3m coint	3m corr	5m coint	5m corr
BTCEUR-BTCGBP	0.5667	0.8758	0.4667	0.8759	0.4667	0.8754
BTCEUR-BTCRUB	0.3333	0.8417	0.3333	0.8417	0.3167	0.8416
BTCEUR-BTCUSD	0.1667	0.9328	0.2000	0.9327	0.2000	0.9329
BTCGBP-BTCRUB	0.3500	0.7606	0.3333	0.7608	0.3333	0.7603
BTCGBP-BTCUSD	0.4833	0.8404	0.4167	0.8403	0.4000	0.8403
BTCRUB-BTCUSD	0.4000	0.8538	0.3333	0.8539	0.3500	0.8543

The transaction cost in the experiment is set to 0.02% commission based on Binance's fee scheme.³ The transaction cost of 0.02% is a flat percentage charge for transactions in both directions. A pair trading leg, including long, the first asset, and short, the second asset, is charged for both long and short actions.

5.1.2. Grid Search and Reinforcement Learning

A grid search is used to find a set of profitable parameters, including the *open/close threshold* and *window size* during the training period (October 2023 to November 2023). Every iteration of the window-sliding pair trading experiments has one set of parameters (window size, open/close threshold) until exhaustion. We start off the exploration based on the experiential estimation of the asset characteristics. The most profitable parameter set will be used to test traditional pair trading during the test period (December 2023) and also for testing the proposed RL strategy.

The profitability in conducting a grid search is measured by the Total Compound Return (RTOT), where V_p and V'_p are the values of the portfolio at the beginning of the period and the end of the period, and t is the total length of the trading period in Equation (11):

$$rtot = (V'_p/V_p)^{1/t} - 1 \times 100\%. \quad (11)$$

During the training period, the most profitable parameter set is *open threshold* = 1.8 z-score, *close threshold* = 0.4 z-score, and *window size* = 900 intervals. Some example results of the grid search are presented in Table 3.

Table 3. Trading parameter tuning.

OPEN _THRES	CLOS_THRES	PERIOD	RTOT (%)
4.0	2.0	2000	0.0651
4.0	0.5	500	0.5024
3.0	1.0	500	0.9993
3.0	0.5	1000	0.8932
3.0	0.5	500	1.0704
2.5	0.3	700	2.1542
2.5	0.5	700	1.5667
3.0	0.3	700	1.3160
2.1	0.4	700	2.5633
2.1	0.3	800	2.6916
2.3	0.4	800	2.3096
2.1	0.4	800	2.8202
2.0	0.4	1000	2.7339
2.0	0.4	900	3.0400
1.9	0.3	900	2.8989
1.9	0.4	900	3.1077
1.8	0.4	900	3.0565
...

The setup of RL-based pair trading relies on these parameters. The window size decides the retrospective length of the spread, and the thresholds decide the zones. Algorithms such as PPO and A2C are applicable to both discrete and continuous action spaces. Some algorithms, e.g., DQN, can only be used in discrete space, and DDPG is only applicable in a continuous space. Therefore, we adopt PPO, DQN, and A2C in RL pair trading, which decide the timing, and PPO, A2C, and SAC in RL pair trading, which decide both the timing and investment quantity. The algorithms are adopted from the Baseline3 collection (Raffin et al. 2021).

5.1.3. Evaluation Metrics

Our main concern is the highest profitability in trading techniques. We care about the cumulative return, which is the total profit for the testing period, as well as the annualized return Compound Annual Growth Rate (CAGR). With $V(t_0)$ as the initial state, $V(t_n)$ as the final state, and $t_n - t_0$ as the period of trading in years, the CAGR is Equation (12):

$$\text{CAGR}(t_0, t_n) = \left(\frac{V(t_n)}{V(t_0)} \right)^{\frac{1}{t_n - t_0}} - 1. \quad (12)$$

There are some popular indicators for distinguishing whether a strategy is profit–risk effective, e.g., the Sharpe ratio. In the Sharpe ratio (Sharpe 1964), R_p is the return of the trading strategy, R_f is the interest rate,⁴ and σ_p is the standard deviation of the portfolio's excess return (Equation (13)):

$$\text{Sharpe Ratio} = \frac{R_p - R_f}{\sigma_p}. \quad (13)$$

We also care about the strategies' activities, such as the order count and win/loss ratio. The indicators used for comparison are presented in Table 4.

Table 4. Descriptive table of evaluation metrics.

Profitability Indicator	Description
Cumulative Return	Profit achieved during trading period
CAGR	Compound Annual Growth Rate
Sharpe Ratio	Risk-adjusted returns ratio
Activity Indicator	Description
Total Action Count	Total orders executed
Win/Loss Action Count	Number of winning/losing trades
Win/Loss Action Ratio	Ratio of winning to losing trades
Max Win/Loss Action	Maximum profit/loss per Action in Bitcoin
Avg Action Profit/Loss	Average profit/loss per trade in Bitcoin
Time in Market	Percentage of time invested in the market
Risk Indicator	Description
Volatility (ann.)	Annualized standard deviation of returns
Skew	Asymmetry of returns distribution
Kurtosis	"Tailedness" of returns distribution

5.2. Experimental Results

We present the profitability and risk results from our experiments along with the trading indicators.

5.2.1. Result Comparison

Our work is compared with standard pair trading (Section 2.1) and state-of-the-art pair trading techniques (Section 3.2).

Our results are presented in Table 5. The results display a positive return for the traditional pair trading technique [Gatev et al. \(2006\)](#). The algorithm A2C displays a positive return for RL pair trading techniques. However, the PPO, SAC, and DQN algorithms do not perform as well as A2C. If we view A2C as the chosen algorithm for pair trading, the results show a steady income from pair trading. The traditional pair trading of [Gatev et al. \(2006\)](#) displays a stable income compared to others due to its rule-based execution stability.

The first adoption of RL₁ pair trading is close to the traditional method. The results table shows that it achieved much better results than the traditional pair trading approach [Gatev et al. \(2006\)](#). The second adoption of RL₂ pair trading is significantly different from RL₁ trading, which decides only timing and produces more profit than other techniques under the same level of volatility. [Kim and Kim's \(2019\)](#) method did not achieve a positive return. Since the method was developed for the forex market, it has not adapted well to the extremely volatile cryptocurrency world.

Table 5. Evaluation metrics comparison between trading techniques.

RL Algo.	Gatev et al. (2006)		Kim and Kim (2019)			RL ₁		RL ₂		
	NA	PPO	A2C	DQN	PPO	A2C	DQN	PPO	A2C	SAC
Profitability										
Cumulative Return	8.33%	−0.16%	−35.16%	−35.79%	1.89%	9.94%	−31.99%	−77.81%	31.53%	−87.12%
CAGR	195.12%	−2.19%	−99.71%	−99.75%	30.05%	278.72%	−99.56%	−100.00%	3974.65%	−100.00%
Sharpe Ratio	25.91	−1.67	−2.04	−2.60	5.44	32.74	−8.77	−1.99	94.34	−1.93
Activities										
Total Action Count	490	43	1248	1062	1304	249	879	3443	229	2798
Won Action Count	284	24	600	503	578	240	232	842	162	917
Lost Action Count	206	19	648	559	726	9	647	2601	67	1881
Win/Loss Action Ratio	1.38	1.26	0.93	0.90	0.80	26.67	0.36	0.32	2.42	0.49
Max Win Action (USD)	75.35	163.52	606.75	606.75	43.72	121.74	70.59	307.78	648.87	160.15
Max Loss Action (USD)	−27.73	−187.86	−763.70	−553.25	−108.51	−21.33	−282.84	−389.22	−64.97	−1456.43
Avg Win Action Profit/Loss (USD)	14.50	41.72	38.68	37.47	5.51	17.77	11.30	15.88	90.94	8.03
Avg Loss Action Profit/Loss (USD)	−2.90	−54.68	−58.75	−60.78	−3.28	−7.06	−24.95	−17.78	−21.00	−23.49
Risk										
Volatility (ann.)	6.01%	3.93%	51.43%	40.43%	3.61%	6.30%	11.92%	53.04%	27.30%	54.66%
Skew	1840	−54	−358	−358	−874	2673	−1899	−374	4314	−3048
Kurtosis	48,145	135	4138	4201	133,603	114,944	54,808	12,987	254,283	138,851

The trading period is from 1 December 2023 to 31 December 2023. The transaction cost is 0.02%, and the interest rate is 5.5%. RL₁ stands for the pair trading that allows Reinforcement Learning to decide upon the investment timing. RL₂ stands for Reinforcement Learning pair trading that allows the RL agent to decide both investment timing and quantity.

Behavior-wise, PPO, DQN, and SAC tend to conduct excessive transactions that are not profitable. On the contrary, A2C have fewer trades but higher profits on each trade. RL₂ pair trading shows further fewer total actions because of the adjusted position action, where we do not consider a position adjustment as one trade until it is closed. Apart from the result in Table 5, the portfolio growth trend with the best-performing RL algorithm agent is presented in Figure 6 (a comparison with Gatev et al. (2006) is provided in Figure A1a in the Appendix). Most of the pair trading experiments, including Gatev et al. (2006), RL₁, and RL₂, display a stable upturn, which is ideal from the perspective of pair trading. From the drawdown graphs, we can observe that RL₁ produces fewer drawdowns compared to the non-RL pair trading method from Gatev et al. (2006) and has a significantly higher win/loss action ratio due to differences in threshold settings. However, RL₁'s cumulative profit is not consistently higher, and when transaction fees are zero, its cumulative profit is slightly lower than that of the Gatev et al. method. RL₂ displays the strongest profitability, despite a lower win/loss action ratio, due to its progressive trading strategy. In general, all three pair trading methods show the ability to generate stable income in a volatile trading market.

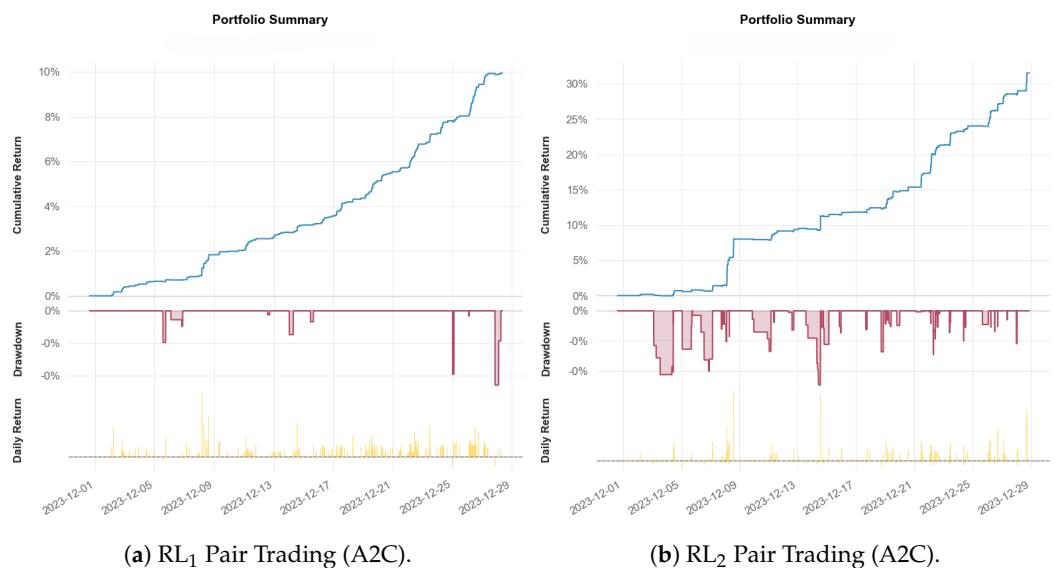


Figure 6. Comparison of portfolio value trends for RL₁ and RL₂ Pair Trading (A2C).

5.2.2. Effect of Transaction Cost

The profitability of high-frequency trading techniques is always significantly impacted by transaction costs. Cryptocurrency exchanges normally provide a large-volume discount scheme. The Binance fee ranges from 0.02% to even 0%, depending on the volume and the holding of their membership token. Considering that the users of these techniques may benefit from different transaction fee tiers, we explore the trading techniques under different transaction cost tiers as well. With more exploration under 0.05%, 0.01%, and 0% transaction costs compared to the default 0.02% transaction cost, we can see the significant impact of decreasing the transaction cost. The participating approaches are adopted with the most profitable algorithm based on the backtesting result (Table 5). We can see that trading techniques generally perform better under lower transaction costs. RL-based techniques tend to perform more trades when the transaction costs are lower.

6. Discussion and Conclusions

Pair trading has been a popular algorithmic trading method for decades. The in-demand high-frequency trading domain requires a fast-track decision-making process. However, the traditional rule-based pair trading technique lacks the flexibility to cater to volatile market movement. In this research, we proposed a mechanism to adopt Reinforcement Learning (RL) to observe the market and produce profitable pair trading decisions.

The first adoption of Reinforcement Learning pair trading grants the RL₁ agent the flexibility to decide action timing. The second adoption of Reinforcement Learning₂ pair trading further gives the RL agent the access to decide the timing and invest quantity.

We compared it to the traditional rule-based pair trading technique (Gatev et al. 2006) and a state-of-the-art RL pair trading technique (Kim and Kim 2019) for December 2023 in the cryptocurrency market for BTCEUR and BTCGBP under a standard future 0.02% transaction cost. Kim and Kim's method does not perform well in the cryptocurrency world. Gatev et al.'s method achieved 8.33% per trading period. Our first adoption of the RL₁ method achieved 9.94%, and the second adoption of the RL₂ method achieved 31.53% returns during the trading period. The outperformance is generally consistent across different transaction costs. The evaluation metrics show that RL-based techniques are generally more active than traditional techniques in the cryptocurrency market under various transaction costs. In general, our trading methods have greater market participation than Gatev et al.'s traditional rule-based pair trading and Kim and Kim's threshold-adaptive RL pair trading (Tables 5 and 6).

Table 6. Evaluation metrics comparison under different transaction costs.

Indicators	Trading Approaches			
0.05% Transaction Fee	Gatev et al. (2006)	Kim and Kim (2019)	RL ₁	RL ₂
Cumulative Profit	5.02%	-0.26%	5.76%	7.40%
Sharpe Ratio	14.60	-2.34	21.00	7.82
Total Action Count	490	43	154	207
Won Action Count	246	23	152	110
Lost Action Count	244	20	2	97
Win/Loss Action Ratio	1.01	1.15	76.00	1.13
Max Win Action (USD)	72.82	114.76	43.36	606.22
Max Loss Action (USD)	-30.26	-169.52	-8.30	-168.81
Avg Win Action Profit/Loss (USD)	13.57	37.94	16.10	70.99
Avg Loss Action Profit/Loss (USD)	-4.99	-47.68	-5.64	-48.28
0.01% Transaction Fee	Gatev et al. (2006)	Kim and Kim (2019)	RL ₁	RL ₂
Cumulative Profit	9.43%	-1.13%	9.88%	33.99%
Sharpe Ratio	29.84	-7.07	33.24	104.40
Total Action Count	490	43	251	181
Won Action Count	317	20	242	149
Lost Action Count	173	23	9	32
Win/Loss Action Ratio	1.83	0.87	26.89	4.66
Max Win Action (USD)	76.20	65.99	121.74	675.23
Max Loss Action (USD)	-26.88	-169.66	-21.33	-27.91
Avg Win Action Profit/Loss (USD)	13.93	24.88	17.52	98.74
Avg Loss Action Profit/Loss (USD)	-2.48	-44.81	-7.06	-10.79
0% Transaction Fee	Gatev et al. (2006)	Kim and Kim (2019)	RL ₁	RL ₂
Cumulative Profit	10.54%	-2.00%	9.94%	80.92%
Sharpe Ratio	33.90	-5.76	32.74	2668.86
Total Action Count	483	43	249	429
Won Action Count	363	23	240	342
Lost Action Count	120	20	9	87
Win/Loss Action Ratio	3.02	1.15	26.67	3.93
Max Win Action (USD)	77.04	163.59	121.74	699.51
Max Loss Action (USD)	-26.03	-217.54	-21.33	-72.21
Avg Win Action Profit/Loss (USD)	13.07	36.69	17.77	104.25
Avg Loss Action Profit/Loss (USD)	-2.43	-80.76	-7.06	-16.68

The trading period is from 1 December 2023 to 31 December 2023 with an interest rate of 5.5%. RL₁ stands for the pair trading that allows Reinforcement Learning to decide upon the investment timing. RL₂ stands for dynamic scaling Reinforcement Learning pair trading that allows the RL agent to decide both investment timing and quantity. We adopted the PPO algorithm from Kim and Kim (2019) and A2C for RL₁ and RL₂.

Comparison between RL-based pair trading revealed the relationship between profitability and actions. Because financial trading is a special case of the RL environment, every action in financial trading is punished by the transaction cost. We notice that profitable RL

trading often has a lower total trade count and higher profit per-win trade. That means the RL is better at spotting chances to make higher profits. RL₂ pair trading produces higher profits because of higher average wins from the position adjustment mechanism. When we adopt the righteous trading method, market volatility and transaction cost play crucial roles in profitable trading. Variable thresholds might not be adaptive to highly volatile markets, and fixed-threshold pair trading could lead to missing trading opportunities. RL with dynamic scaling investment could be a good direction in volatile market conditions if low transaction costs are achievable.

The techniques presented have certain limitations and offer opportunities for future work. One limitation is the relatively limited dataset scope, which could be expanded to include more diverse assets and longer timeframes to improve generalization. Additionally, focusing only on two-leg strategies restricts the potential for complex arbitrage opportunities; expanding to multi-leg strategies would enhance robustness. The computational demand during training can also be resource-intensive, requiring system parameter tuning. The model lacks consideration for transaction costs, which might impact real-world profitability. A lack of direct comparison with traditional models is another shortcoming. Future work could involve developing the Reinforcement Learning (RL) approach to multi-leg strategies, integrating pair formation into the trading process, cross-validating across different environments, and experimenting with alternative reward functions to improve decision-making and risk management.

Author Contributions: Conceptualization, A.M. and H.Y.; methodology, A.M. and H.Y.; software, A.M. and H.Y.; validation, H.Y.; formal analysis, H.Y.; investigation, H.Y.; resources, A.M. and H.Y.; data curation, H.Y.; writing—original draft preparation, H.Y.; writing—review and editing, A.M.; visualization, H.Y.; supervision, A.M.; project administration, A.M.; funding acquisition, Not applicable. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding

Institutional Review Board Statement: Not applicable.

Data Availability Statement: The original data presented in the study are openly available from Binance Exchange accessed on 8 November 2024 at (<https://data.binance.vision/>).

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A

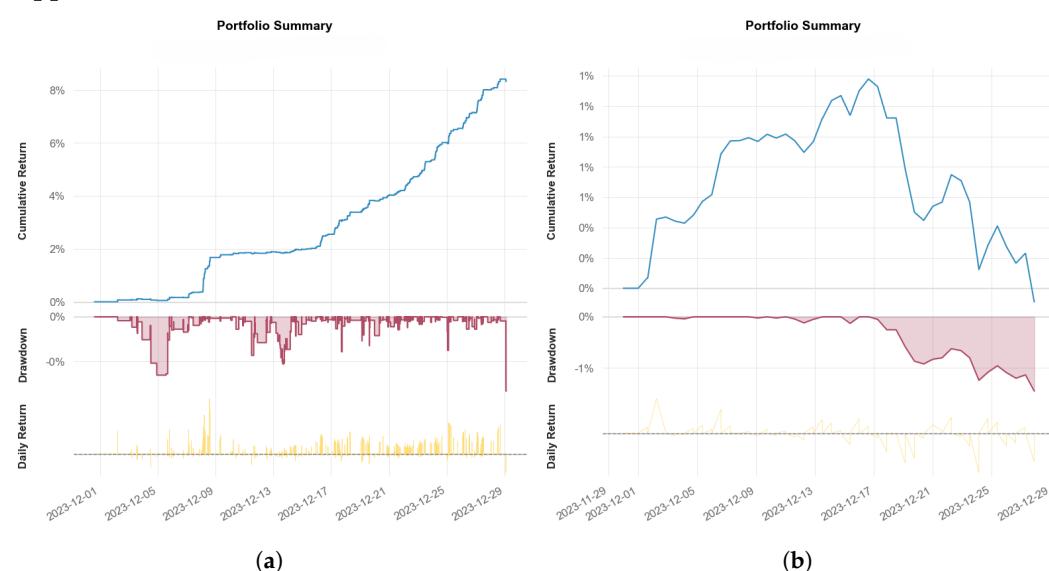


Figure A1. Comparison of Pair Trading strategies from (a) [Gatev et al. \(2006\)](#) and (b) [Kim and Kim \(2019\)](#).

Notes

- 1 <https://www.binance.com/en>, accessed on 8 November 2024.
- 2 While calculating the co-integration and correlation, intervals with low volume trades are exempted from the calculation.
- 3 <https://www.binance.com/en/fee/futureFee>, accessed on 8 November 2024.
- 4 We adopt the Federal Reserve interest rate of 5.5%, which is correct as of 13 June 2024.

References

- AlMahamid, Fadi, and Katarina Grolinger. 2021. Reinforcement learning algorithms: An overview and classification. Paper presented at the 2021 IEEE Canadian Conference on Electrical and Computer Engineering (CCECE), Virtual, September 12–17; pp. 1–7.
- Bellman, Richard. 1957. A Markovian Decision Process. *Journal of Mathematics and Mechanics* 6: 679–84. [CrossRef]
- Bertoluzzo, Francesco, and Marco Corazza. 2007. Making Financial Trading by Recurrent Reinforcement Learning. In *Knowledge-Based Intelligent Information and Engineering Systems*. Edited by Bruno Apolloni, Robert J. Howlett and Lakhmi Jain. Lecture Notes in Computer Science. Berlin and Heidelberg: Springer, pp. 619–26. [CrossRef]
- Brogaard, Jonathan, Terrence Hendershott, and Ryan Riordan. 2014. High-Frequency Trading and Price Discovery. *The Review of Financial Studies* 27: 2267–306. [CrossRef]
- Burgess, A. Neil. 2003. Using Cointegration to Hedge and Trade International Equities. In *Applied Quantitative Methods for Trading and Investment*. Hoboken: John Wiley & Sons, Ltd., pp. 41–69. [CrossRef]
- Dickey, David A., and Wayne A. Fuller. 1979. Distribution of the Estimators for Autoregressive Time Series with a Unit Root. *Journal of the American Statistical Association* 74: 427–31. [CrossRef]
- Do, Binh, and Robert Faff. 2010. Does Simple Pairs Trading Still Work? *Financial Analysts Journal* 66: 83–95. [CrossRef]
- Dunis, Christian L., and Richard Ho. 2005. Cointegration portfolios of European equities for index tracking and market neutral strategies. *Journal of Asset Management* 6: 33–52. [CrossRef]
- Dybvig, Philip H., and Stephen A. Ross. 1989. Arbitrage. In *Finance*. Edited by John Eatwell, Murray Milgate and Peter Newman. London: Palgrave Macmillan UK, pp. 57–71. [CrossRef]
- Fadok, David S., John Boyd, and John Warden. 1995. Air power's quest for strategic paralysis. *Proceedings of the School of Advanced Airpower Studies*. Available online: https://media.defense.gov/2017/Dec/27/2001861508/-1/-1/0/T_0029_FADOK_BOYD_AND_WARDEN.PDF (accessed on 8 November 2024)
- Gatev, Evan, William N. Goetzmann, and K. Geert Rouwenhorst. 2006. Pairs Trading: Performance of a Relative Value Arbitrage Rule. *The Review of Financial Studies* 19: 797–827. [CrossRef]
- Gold, Carl. 2003. FX trading via recurrent reinforcement learning. Paper presented at the 2003 IEEE International Conference on Computational Intelligence for Financial Engineering, Hong Kong, China, March 20–23, pp. 363–70, ISBN 9780780376540. [CrossRef]
- Haarnoja, Tuomas, Aurick Zhou, Kristian Hartikainen, George Tucker, Sehoon Ha, Jie Tan, Vikash Kumar, Henry Zhu, Abhishek Gupta, Pieter Abbeel, and et al. 2019. Soft Actor-Critic Algorithms and Applications. *arXiv arXiv:1812.05905*. [CrossRef]
- Han, Weiguang, Jimin Huang, Qianqian Xie, Boyi Zhang, Yanzhao Lai, and Min Peng. 2023. Mastering Pair Trading with Risk-Aware Recurrent Reinforcement Learning. *arXiv arXiv:2304.00364*.
- Huang, Chien Yi. 2018. Financial Trading as a Game: A Deep Reinforcement Learning Approach. *arXiv arXiv:1807.02787*. [CrossRef]
- Huck, Nicolas. 2010. Pairs trading and outranking: The multi-step-ahead forecasting case. *European Journal of Operational Research* 207: 1702–16. [CrossRef]
- Kim, Taewook, and Ha Young Kim. 2019. Optimizing the Pairs-Trading Strategy Using Deep Reinforcement Learning with Trading and Stop-Loss Boundaries. *Complexity* 2019: e3582516. [CrossRef]
- Liu, Xiao-Yang, Hongyang Yang, Jiechao Gao, and Christina Dan Wang. 2021. FinRL: Deep Reinforcement Learning Framework to Automate Trading in Quantitative Finance. Paper presented at the Proceedings of the Second ACM International Conference on AI in Finance, Virtual Event, November 3–5, pp. 1–9. [CrossRef]
- Liu, Xiao-Yang, Hongyang Yang, Qian Chen, Runjia Zhang, Liuqing Yang, Bowen Xiao, and Christina Dan Wang. 2022a. FinRL: A Deep Reinforcement Learning Library for Automated Stock Trading in Quantitative Finance. *arXiv arXiv:2011.09607*. [CrossRef]
- Liu, Xiao-Yang, Ziyi Xia, Jingyang Rui, Jiechao Gao, Hongyang Yang, Ming Zhu, Christina Dan Wang, Zhaoran Wang, and Jian Guo. 2022b. FinRL-Meta: Market Environments and Benchmarks for Data-Driven Financial Reinforcement Learning. *arXiv arXiv:2211.03107*. [CrossRef]
- Lucarelli, Giorgio, and Matteo Borrotti. 2019. A Deep Reinforcement Learning Approach for Automated Cryptocurrency Trading. In *Artificial Intelligence Applications and Innovations*. IFIP Advances in Information and Communication Technology. Edited by John MacIntyre, Ilias Maglogiannis, Lazaros Iliadis and Elias Pimenidis. Cham: Springer International Publishing, pp. 247–58. [CrossRef]
- Mandelbrot, Benoit. 1967. The Variation of Some Other Speculative Prices. *The Journal of Business* 40: 393–413. [CrossRef]
- Maringer, Dietmar, and Tikes Ramtohul. 2012. Regime-switching recurrent reinforcement learning for investment decision making. *Computational Management Science* 9: 89–107. [CrossRef]
- Meng, Terry Lingze, and Matloob Khushi. 2019. Reinforcement Learning in Financial Markets. *Data* 4: 110. [CrossRef]

- Mnih, Volodymyr, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. 2013. Playing Atari with Deep Reinforcement Learning. *arXiv* arXiv:1312.5602. [[CrossRef](#)]
- Mohammadshafie, Alireza, Akram Mirzaeinia, Haseebullah Jumakhan, and Amir Mirzaeinia. 2024. Deep Reinforcement Learning Strategies in Finance: Insights into Asset Holding, Trading Behavior, and Purchase Diversity. *arXiv* arXiv:2407.09557. [[CrossRef](#)]
- Perlin, Marcelo. 2007. M of a Kind: A Multivariate Approach at Pairs Trading. Available online: <https://doi.org/10.2139/ssrn.952782> (accessed on 8 November 2024).
- Perlin, Marcelo Scherer. 2009. Evaluation of pairs-trading strategy at the Brazilian financial market. *Journal of Derivatives & Hedge Funds* 15: 122–36. [[CrossRef](#)]
- Pricope, Tidor-Vlad. 2021. Deep Reinforcement Learning in Quantitative Algorithmic Trading: A Review. *arXiv* arXiv:2106.00123. [[CrossRef](#)]
- Raffin, Antonin, Ashley Hill, Adam Gleave, Anssi Kanervisto, Maximilian Ernestus, and Noah Dormann. 2021. Stable-baselines3: Reliable reinforcement learning implementations. *Journal of Machine Learning Research* 22: 1–8.
- Sarmento, Simão Moraes, and Nuno Horta. 2020. Enhancing a Pairs Trading strategy with the application of Machine Learning. *Expert Systems with Applications* 158: 113490. [[CrossRef](#)]
- Schulman, John, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal Policy Optimization Algorithms. *arXiv* arXiv:1707.06347. [[CrossRef](#)]
- Sharpe, William F. 1964. Capital Asset Prices: A Theory of Market Equilibrium Under Conditions of Risk. *The Journal of Finance* 19: 425–42. [[CrossRef](#)]
- Silver, David, and Demis Hassabis. 2016. AlphaGo: Mastering the ancient game of Go with Machine Learning. Available online: <https://research.google/blog/alphago-mastering-the-ancient-game-of-go-with-machine-learning/> (accessed on 8 November 2024).
- Sutton, Richard S., and Andrew G. Barto. 2018. *Reinforcement Learning: An Introduction*. Cambridge: MIT Press.
- Vergara, Gabriel, and Werner Kristjanpoller. 2024. Deep reinforcement learning applied to statistical arbitrage investment strategy on cryptomarket. *Applied Soft Computing* 153: 111255. [[CrossRef](#)]
- Wang, Cheng, Patrik Sandås, and Peter Beling. 2021. Improving Pairs Trading Strategies via Reinforcement Learning. Paper Presented at the 2021 International Conference on Applied Artificial Intelligence (ICAPAI), Halden, Norway, May 19–21, pp. 1–7. [[CrossRef](#)]
- Yang, Hongshen, and Avinash Malik. 2024. Optimal market-neutral currency trading on the cryptocurrency platform. *arXiv* arXiv:2405.15461. [[CrossRef](#)]
- Zhang, Jin, and Dietmar Maringer. 2016. Using a Genetic Algorithm to Improve Recurrent Reinforcement Learning for Equity Trading. *Computational Economics* 47: 551–67. [[CrossRef](#)]
- Zhang, Zihao, Stefan Zohren, and Roberts Stephen. 2020. Deep Reinforcement Learning for Trading. *The Journal of Financial Data Science* 2: 25–40. [[CrossRef](#)]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.