Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai

Improved pairs trading strategy using two-level reinforcement learning framework

Zhizhao Xu [a], Chao Luo [a,b,*]

[a] School of Information Science and Engineering, Shandong Normal University, Jinan, 250014, China
[b] Shandong Provincial Key Laboratory for Novel Distributed Computer Software Technology, Jinan, 250014, China

ARTICLE INFO

Keywords:
Pair trading
Reinforcement learning
Time series
Deep learning

ABSTRACT

Pairs trading is a popular classic neutral trading strategy in financial market. Deep reinforcement learning (DRL) has been widely used to improve the performance of this strategy. However, most works primarily focused on setting trading signals, but ignored selecting appropriate trading pairs. In this paper, a novel two-level reinforcement learning framework is proposed, where both pair selection and trading thresholds setting are involved. For pair selection, an Extended Option-Critic (EOC) method is utilized, which allows the agent to select trading pair on non-fixed length of time intervals. For trading thresholds setting, a three-agent Multi-Agent Deep Deterministic Policy Gradient (MADDPG) method is used for setting the opening and stop-loss thresholds as well as decide whether to trade. The simulation results in the Chinese futures market demonstrate that our proposed method achieves higher returns compared to traditional methods and popular reinforcement learning approaches.

1. Introduction

Pairs Trading is a common statistical arbitrage strategy that focuses on spreads between two price series. This strategy is based on the mean-reversion property of the market, which states that if two or more price series have similar trends, their spreads will remain within a reasonable range of movement (Gatev et al., 2006). When the spread deviates from the mean, the market mechanism will eventually reverse the trend and bring the spread back to its mean level. By taking advantage of this mean-reversion property, pairs trading can be used to open positions when the spread is out of balance and close them when the spread returns to the mean, allowing for arbitrage opportunities (Gatev et al., 2006; Vidyamurthy, 2004; Elliott et al., 2005; Leung and Li, 2013). In summary, pairs trading involves offsetting risk by taking positions in two different futures and is a market-neutral trading strategy with good hedging ability.

Machine learning methods have achieved remarkable results in modelling of complex systems (Pozna et al., 2012; Pozna and Precup, 2012; Hedrea et al., 2021; Precup et al., 2022) and classification (Şeref et al., 2017; Panagopoulos et al., 2019). Researchers have proposed various methods utilizing machine learning to enhance pairs trading strategies, with a particular focus on employing Deep Reinforcement Learning (DRL) methods. DRL is a popular branch of deep learning that has achieved impressive success in recent years in areas such as gaming, robot control, parametric optimization, and machine vision (Mnih et al., 2013; Lillicrap et al., 2016; Fujimoto et al., 2018). In DRL, an agent interacts with the environment to maximize the total reward. Due to the complexity of financial markets and the rapid nature of high-frequency trading, trading opportunities are often short-lived, making it impossible for human traders to keep pace with market

movements and make trading decisions quickly. DRL can automate trades by capturing potential trading opportunities and performing feature extraction instead of relying on human traders, and well-designed agents have the potential to outperform experienced human traders. In algorithmic trading, DRL combines the perceptual capabilities of deep learning with the decision-making capabilities of reinforcement interacts with the market to maximize total returns (Deng et al., 2016; Lin et al., 2022; Shavandi and Khedmati, 2022). There are currently many studies in this area, such as Kim and Kim (2019), Lu et al. (2022) and Kim et al. (2022).

However, while these studies address the limitations of traditional pairs trading strategies in trading signal setting, they typically still rely on traditional methods for pair selection. This can lead to unnecessary losses, for example, when the cointegration between the pair dissipates,

[*] Corresponding author. School of Information Science and Engineering, Shandong Normal University, Jinan, 250014, China.
E-mail address: cluo79@gmail.com (C. Luo).

https://doi.org/10.1016/j.engappai.2023.107148
Received 13 March 2023; Received in revised form 8 August 2023; Accepted 11 September 2023
Available online 22 September 2023

making it difficult to profit from the spread reverting to normal levels. Additionally, some studies attempt to use a single agent to simultaneously select opening and stop-loss thresholds, which are often separated by a fixed interval. Our first goal is to improve pairs trading strategies by using DRL methods to improve both pair selection and trading threshold setting: dynamic pair selection during pair trading allows us to continuously trade highly profitable pairs, and optimizing the opening and stop-loss thresholds for each selected pair to maximize returns. To build on these studies, we propose a Multi-Agent Deep Deterministic Policy Gradient (MADDPG, Lowe et al., 2017) approach that set opening and stop-loss thresholds and decides when to trade separately. This approach emphasizes communication among agents, allowing multiple agents to learn appropriate strategies for their own tasks and maximize returns by communicating and cooperating with each other.

In terms of pair selection, apart from using the cointegration method (Vidyamurthy, 2004; Elliott et al., 2005; Leung and Li, 2013; Galenko et al., 2012; Lin et al., 2006; Bertram, 2010) and distance method (Gatev et al., 2006; Pole, 2011; Do and Faff, 2012; Chen et al., 2019), research on stock selection (Winkel et al., 2022; Wang et al., 2019; Li et al., 2022; Zha et al., 2022) in portfolio management is similar to the problem addressed in this paper. These approaches conduct stock selection at fixed time intervals, which are usually manually defined, such as one calendar month or a specified time window length. There are few discussions on using DRL methods to address the issue of selecting trading pairs at non-fixed time intervals. This may result in additional losses since it is difficult to determine an exact time interval length with high profitability for assets, and the time interval may not have a fixed length. Unlike these works that design network structures to extract features at fixed time intervals, our second objective is to design a DRL method for the pair selection model in the two-level framework. We proposed Extended Option-Critic (EOC) method, which is an extension of the Option-Critic (OC) architecture (Bacon et al., 2017). This method can automatically determine the appropriate time interval length and select profitable trading pairs to maximize returns.

In this paper, a novel two-level reinforcement learning framework is proposed for pairs trading which takes into account both pair selection (PS for short) and trade thresholds setting (TS for short). The approach utilizes two innovative methods, namely an EOC method for PS (PS-EOC) and a MADDPG method for TS (TS-MADDPG). PS-EOC enable pair selection with non-fixed length of time intervals by learning policy over options and termination functions. TS-MADDPG utilizes three agents to set the opening and stop-loss thresholds for trading pairs, as well as determining whether to execute trades. Simulations in the Chinese futures market validate the effectiveness of the proposed framework. The contributions of this work can be summarized as follows:

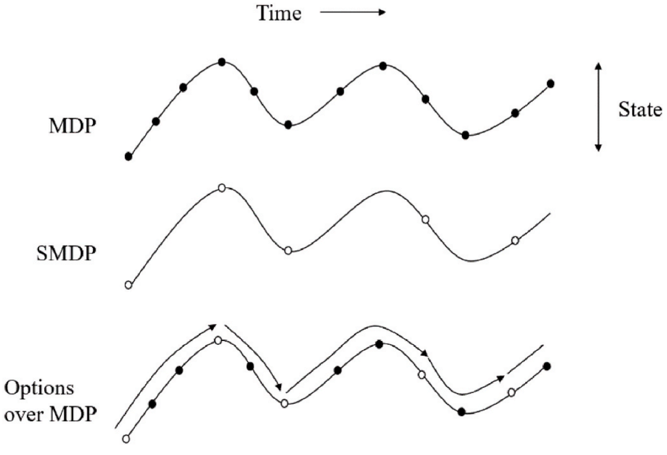


Fig. 1. MDP, SMDP and options over MDP (Sutton et al., 1999).

- A novel two-level reinforcement learning framework is proposed, which addresses both pair selection and trading thresholds setting to improve the profitability of pairs trading strategy. By comparing with other methods in Chinese futures market, the efficiency of the two-level framework is demonstrated.
- Proposed an Extended Option-Critic method, which extends the Option-Critic architecture to enable pair selection on non-fixed length of time intervals by learning policy over options and termination function, improved the flexibility of pairs trading strategy.
- Applying the MADDPG method with three agents to set the opening and stop-loss thresholds as well as the decision of whether to trade separately through communication and collaboration among multiple agents.

This paper is organized as follows: section II introduces the related works, section III presents our proposed two-level reinforcement framework, section IV describes the simulation setup as well as the simulation results, and section V gives a summary and outlook.

2. Related works

2.1. DRL for pairs trading

Traditional pairs trading strategies generally involve two steps: pair selection and trading signal setting. For the pair selection, common methods include the cointegration approach (Vidyamurthy, 2004; Elliott et al., 2005; Leung and Li, 2013; Lin et al., 2006; Bertram, 2010; Galenko et al., 2012) and the distance-based approach (Gatev et al., 2006; Pole, 2011; Do and Faff, 2012; Chen et al., 2019). The distance-based approach measures the distance between asset price sequences, such as using the Euclidean squared distance method, and selects the closest two assets to form a trading pair. The cointegration approach, based on the Engle-Granger cointegration test (Elliott et al., 2005), selects trading pairs consisting of assets with long-term stationarity in their spreads. These methods typically involve selecting trading pairs based on historical data, with the pairs remaining unchanged during the trading process.

Research on stock selection in portfolio management is similar to the problem addressed in this paper (Winkel et al., 2022; Wang et al., 2019; Li et al., 2022; Zha et al., 2022). Wang et al. (2019) proposed a DRL method based on temporal abstraction for automatic stock selection and setting of trading weights in portfolio management. They first extract features from the time series of individual assets using Long short-term memory (LSTM, Hochreiter and Schmidhuber, 1997) with history state attention, and then model the relationships among assets using attention mechanisms. Finally, investment proportions for each asset are determined based on the winner scores outputted by the attention network. Zha et al. (2022) proposed a hierarchical reinforcement learning framework for stock selection and portfolio management. In this framework, the high-level agent is responsible for selecting stocks with a high probability of profitability, while the lower-level agent performs portfolio optimization for greater profitability. Unlike these works that design network structures to extract features at fixed time intervals, our proposed EOC method improves returns by selecting trading pairs at non-fixed time intervals.

There are two main approaches for trading signal setting using DRL: direct determination of trading behavior by the agent (Brim, 2020; Fallahpour et al., 2016; Wang et al., 2021; Sarmento and Horta, 2020) and indirect determination of trading behavior through the setting of opening and stop loss thresholds (Kim, T., and Kim, H. Y. 2019; Lu et al., 2022; Kim et al., 2022). Kim and Kim (2019) used Deep Q-Learning Network (DQN, Mnih et al., 2013) to set the opening and stop-loss thresholds for pairs trading strategies. Lu et al. (2022) built on Kim and Kim (2019) by adding a structural break-aware mechanism that predicts the probability of future structure breaks, allowing the agent to make advance risk control. Kim et al. (2022) proposed a hybrid DRL

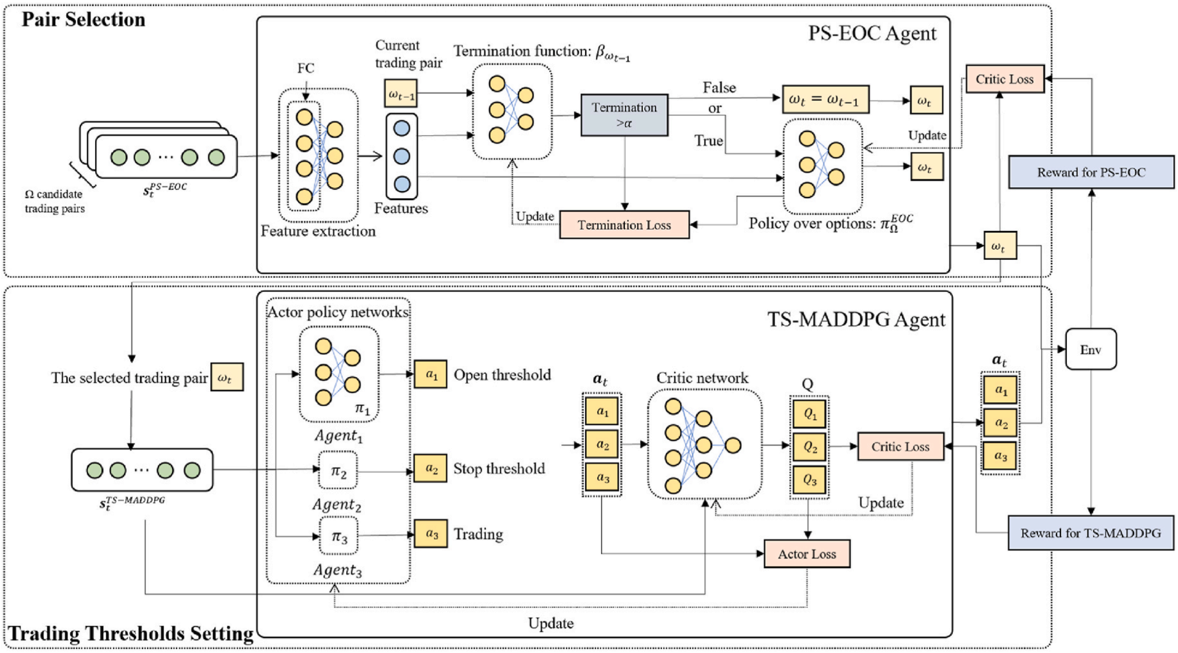


Fig. 2. General structure of the two-level reinforcement learning framework.

framework that uses Twin Delayed Deep Deterministic Policy Gradient (TD3, Fujimoto et al., 2018) to directly determine trading behavior and Double Deep Q-Learning Network (DDQN, Van Hasselt et al., 2016) to determine stop-loss thresholds for pairs trading strategies. They also use techniques such as gate structure, clustering, and dimensionality reduction to adaptively extract features. Building upon these works, we apply the MADDPG method for trading thresholds setting.

2.2. The options framework

The options framework was firstly proposed in Sutton et al. (1999). Bacon et al. (2017) derived the policy gradient theorem for options and introduced an OC architecture that can learn the internal policies and the termination conditions of options, in tandem with the policy over options. Unlike Markov Decision Process (MDP, Puterman, 2014), reinforcement learning methods based on options follow the SMDP (Sutton et al., 1999; Bacon et al., 2017; Precup, 2000), which describes a process where the time intervals between decision points are not necessarily the same in continuous and discrete time. The following Fig. 1 from Sutton et al. (1999) provides an example of the MDP, SMDP, and options over MDP.

3. Model

We propose a two-level reinforcement learning framework to improve the pairs trading strategy in terms of both Pair Selection and Trading Thresholds Setting. The general structure of the model is illustrated in Fig. 2.

3.1. Two-level reinforcement learning framework

As shown in Fig. 2, the two-level reinforcement learning framework consists of two parts. The upper part illustrates the process of using EOC for pair selection, while the lower part demonstrates the process of using MADDPG for trading threshold setting. For the agents, the reinforcement learning environment is the market. Assuming there are Ω candidate trading pairs, the close prices of the two contracts forming pair ω are denoted as $p_{1,t}^{\omega}$ and $p_{2,t}^{\omega}$, the price of the spread is p_t^{ω}, $p_t^{\omega} = p_{1,t}^{\omega} - p_{2,t}^{\omega}$. At time step $t = 0$, all trading pairs have a historical data window of length W. We calculate the $spread_t^{\omega}$ at time step t as follows:

$$spread_t^{\omega} = \frac{p_t^{\omega} - mean\left(p_{t-W}^{\omega}, \ldots, p_t^{\omega}\right)}{std\left(f\left(p_{t-W}^{\omega}, \ldots, p_t^{\omega}\right)\right)} \tag{1}$$

where *mean* is the function for calculating the mean of the sequence, *std* is the function for calculating the standard deviation of the sequence, and f is the function for calculating the decentralized sequence.

At each time step t, PS-EOC agent receives the current state $\boldsymbol{s}_t^{PS-EOC}$ and the termination function $\beta_{\omega_{t-1}}$ determines whether the previous trading pair ω_{t-1} needs to be closed. If $\beta_{\omega_{t-1}}(\boldsymbol{s}_t^{PS-EOC}) > \alpha$, the agent selects a new pair ω_t to trade based on the policy over options π_Ω^{EOC}, i.e., $\omega_t = \pi_\Omega^{EOC}(\boldsymbol{s}_t^{PS-EOC})$; otherwise, the pair remains unchanged, i.e., $\omega_t = \omega_{t-1}$. α is a hyperparameter which is the confidence threshold for the termination probability.

Next, a three-agent TS-MADDPG method with $\pi_1^{\omega_t}, \pi_2^{\omega_t}, \pi_3^{\omega_t}$ corresponding to pair ω_t requires selecting appropriate opening threshold (ol) and stop-loss threshold (sl) based on the current state, and deciding whether to trade at the current time point, i.e., $\boldsymbol{a}_t = (ol, sl, Trading) = (\pi_1^{\omega_t}(\boldsymbol{s}_t^{TS-MADDPG}),\ \pi_2^{\omega_t}(\boldsymbol{s}_t^{TS-MADDPG}),\ \pi_3^{\omega_t}(\boldsymbol{s}_t^{TS-MADDPG}))$, when $Trading = True$, the pair trading strategy is executed based on the current spread of the trading pair $spread_t^{\omega}$:

Open: If there is no current position and $ol < abs(spread_t^{\omega}) < sl$, a short position is taken when $spread_t^{\omega} > 0$, and a long position is taken when $spread_t^{\omega} < 0$.

Close: If there is a current short position and $spread_t^{\omega} < 0$, a long position is taken to close the position. If there is a current long position and $spread_t^{\omega} > 0$, a short position is taken to close the position.

Stop-loss: If there is a current short position and $spread_t^{\omega} > sl$, a long position is taken to stop the loss. If there is a current long position and $spread_t^{\omega} < -sl$, a short position is taken to stop the loss.

There are transaction costs c associated with each trade, the calculation method is provided in Section 4.1. After close or stop-loss, the $return_t$ is calculated as follows:

$$return_t = p_{t_{short}}^{\omega} - p_{t_{long}}^{\omega} - 2c \tag{2}$$

Algorithm 1 outlines the complete process of improving pairs trading strategy using the two-level framework.

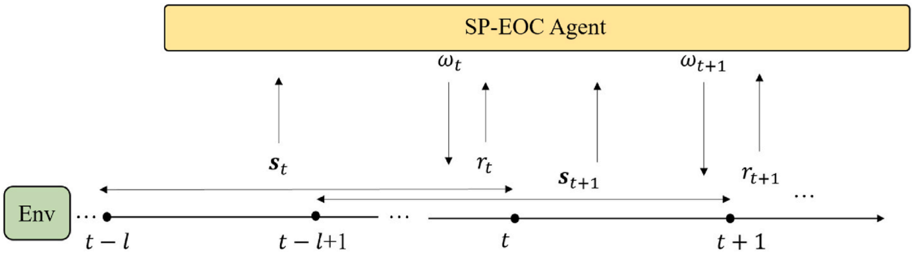


Fig. 3. Schematic diagram of PS-EOC.

Algorithm 1. Improved Pairs Trading Strategy Using Two-Level Framework

```
Initialize $\pi_\Omega^{EOC}, \beta_\omega, \pi_1^\omega, \pi_2^\omega, \pi_3^\omega$
$step = 0, minsteps \leftarrow L$
for $t = 1, T$ do
    Observe $\boldsymbol{s}_t^{PS-EOC}$
    if $t = 1$ or $step > minsteps$ then
        $step = 0$
        if $\beta_{\omega_{t-1}}(\boldsymbol{s}_t^{PS-EOC}) > \alpha$ then
            $\omega_t = \pi_\Omega^{EOC}(\boldsymbol{s}_t^{PS-EOC})$
        else
            $\omega_t = \omega_{t-1}$
    Observe $\boldsymbol{s}_t^{TS-MADDPG}$ on trading pair $\omega_t$
    $\boldsymbol{a}_t = \left(\pi_1^{\omega_t}(\boldsymbol{s}_t^{TS-MADDPG}), \pi_2^{\omega_t}(\boldsymbol{s}_t^{TS-MADDPG}), \pi_3^{\omega_t}(\boldsymbol{s}_t^{TS-MADDPG})\right)$
    Execute pairs trading strategy on trading pair $\omega_t$ with $(ol, sl, Trading) = \boldsymbol{a}_t$
    Calculate $return_t$ by Eq (2)
    $step += 1$
end for
```

In Algorithm 1, *minsteps* is a hyperparameter to prevent the pairs from switching too often to cause losses. Our goal is to maximize the cumulative return over all time points by selecting appropriate trading pairs at non-fixed length intervals and setting suitable trading thresholds for the trading pairs.

In the following section, we will provide detailed description of PS-EOC and TS-MADDPG, including their environment settings.

3.2. PS-EOC

In practical trading scenarios, it may not be possible to determine the optimal length of a trading interval for each pair in advance. To address this issue, we propose an extension of the OC architecture namely EOC, which allow agent selecting trading pairs at non-fixed length intervals. We refer to this method as PS-EOC and the framework is illustrated in Fig. 3.

Fig. 3 shows that at each time step t, the agent receives the current state and the termination function β_{ω_t} determines whether the previous trading pair needs to be closed and a new pair needs to be selected based on the policy over options. If the termination condition is met, the agent selects a new pair to trade based on the policy over options; otherwise, the agent continues trading the current pair. The key challenge here is to learn an effective termination function, which we address by extending the OC architecture.

The OC architecture is a framework for learning a set of option policies over base actions that enables the agent to solve problems in different situations using different options. The OC architecture produces an option by estimating the value function of the options. Let the policy over options be denoted as π_ω, and its parameters be θ. Then the option value function is defined as:

$$Q_\Omega(\boldsymbol{s},\omega) = \sum_a \pi_{\omega,\theta}(a|\boldsymbol{s}) Q_U(\boldsymbol{s},\omega,a) \tag{3}$$

where $Q_U : \boldsymbol{S} \times \boldsymbol{\Omega} \times \boldsymbol{A} \to \boldsymbol{R}$ is the value of the action performed in the context of the state-option pair. It is defined as:

$$Q_U(\boldsymbol{s},\omega,a) = r(\boldsymbol{s},a) + \gamma \sum_{s'} P(\boldsymbol{s}'|\boldsymbol{s},a) U(\omega,\boldsymbol{s}') \tag{4}$$

In Eq (4), γ is the discount factor, $\boldsymbol{P}$ is the state transfer probability function, and the function $U : \boldsymbol{\Omega} \times \boldsymbol{S} \to \mathbb{R}$ is called the option-value function upon arrival. Note that $(\boldsymbol{s},\omega)$ leads to the expanded state space. However, the OC architecture does not use this space explicitly. Let the termination function β_ω be the argument of ϑ, the value of execution ω at entry into state s' is given by the following equation:

$$U(\omega,s') = \left(1-\beta_{\omega,\vartheta}(\boldsymbol{s}')\right) Q_\Omega(\boldsymbol{s}',\omega) + \beta_{\omega,\vartheta}(\boldsymbol{s}') V_\Omega(\boldsymbol{s}') \tag{5}$$

If the option ω_t has been started or is being executed at the state $\boldsymbol{s}_t$, the probability of transitioning in one step to $(\boldsymbol{s}_{t+1},\omega_{t+1})$ with the probability of:

$$P(\boldsymbol{s}_{t+1}, \omega_{t+1} | \boldsymbol{s}_t, \omega_t) = \sum_a \pi_{\omega_t,\theta}(a|\boldsymbol{s}) P(\boldsymbol{s}_{t+1}|\boldsymbol{s}_t, a)\left(1 - \beta_{\omega_t,\vartheta}(\boldsymbol{s}_{t+1})\right) 1_{\omega_t=\omega_{t+1}}$$
$$+ \beta_{\omega_t,\vartheta}(\boldsymbol{s}_{t+1})\pi_\Omega(\omega_{t+1}|\boldsymbol{s}_{t+1}) \tag{6}$$

We want to explicitly use the expanded state space (s, ω) to allow policy over options can directly interact with the environment. When the internal policy is determined, the Q_Ω is a function that depends solely on ϑ. Using Eqs (3) and (4), the option-value function of EOC is:

$$Q_\Omega^{EOC}(\boldsymbol{s}, \omega) = r(\boldsymbol{s}, \omega) + \gamma \sum_{s'} P(s'|\boldsymbol{s}, \omega) U(\omega, s') \tag{7}$$

Here, $r(\boldsymbol{s}, \omega)$ is the reward obtained for executing the strategy corresponding to ω. $U(\omega, s')$ remains the same:

$$U(\omega, s') = \left(1 - \beta_{\omega,\vartheta}(s')\right) Q_\Omega(s', \omega) + \beta_{\omega,\vartheta}(s') V_\Omega(s') \tag{8}$$

Assuming π_Ω^{EOC} is a greedy policy for option, the corresponding one-step policy update goal from Eq (4) as g_t:

$$g_t = r_{t+1} + \gamma\left(\left(1 - \beta_{\omega_t,\vartheta}(\boldsymbol{s}_{t+1})\right) Q_\Omega^{EOC}(\boldsymbol{s}_{t+1}, \omega) + \beta_{\omega_t,\vartheta}(\boldsymbol{s}_{t+1}) \max_\omega Q_\Omega^{EOC}(\boldsymbol{s}_{t+1}, \omega)\right) \tag{9}$$

With this, it is possible to give the loss function of the policy over options network π_Ω^{EOC}:

$$Loss_{\pi_\Omega^{EOC}} = \mathbb{E}\left(\left(Q_\Omega^{EOC}(\boldsymbol{s}_t, \omega_t) - g_t\right)^2\right) \tag{10}$$

The loss function of termination function $\beta_{\omega,\vartheta}$:

$$Loss_{\beta_\omega} = \beta_\omega(\boldsymbol{s})\left(Q_\Omega^{EOC}(\boldsymbol{s}, \omega) - \max_\omega Q_\Omega^{EOC}(\boldsymbol{s}, \omega) + \eta\right) \tag{11}$$

where η is a correction factor to prevent the output from converging to the same option or the termination function from failing. Algorithm 2 outlines the training process of PS-EOC.

Algorithm 2. Training process of PS-EOC

```
1:  Initialize replay memory $D$
2:  Initialize $\pi_\Omega^{EOC}$ and $\beta_\omega$
3:  for $t = 1, T$ do
4:      Observe $\boldsymbol{s}_t$
5:      if $Bernoulli(\beta_{\omega_{t-1}}(\boldsymbol{s}_t)) = True$ then
6:          With probability $\epsilon$ select a random $\omega_t$
7:          Otherwise $\omega_t = \pi_\Omega^{EOC}(\boldsymbol{s}_t)$
8:      else
9:          $\omega_t = \omega_{t-1}$
10:     Execute pairs trading strategy on $\omega_t$, obtain $r_t$ and observe $\boldsymbol{s}_{t+1}$
11:     Store transition $(\boldsymbol{s}_t, \omega_t, r_t, \boldsymbol{s}_{t+1})$ in $D$
12:     Sample random minibatch $(\boldsymbol{s}_j, \omega_j, r_j, \boldsymbol{s}_{j+1})$ from $D$
13:     Calculate $g_j$ by Eq (9)
14:     Update $\pi_\Omega^{EOC}$ by minimize Eq (10)
15:     Update $\beta_{\omega_j}$ by minimize Eq (11)
16: end for
```

In Algorithm 2, $\boldsymbol{s}^{PS-EOC}$ is short as $\boldsymbol{s}$. We use Bernoulli distribution sampling and ϵ-greedy policy to explore the environment at Step 5 and Step 6. Step 10 can involve implementing a pairs trading strategy with either static trading thresholds or TS methods. For example, to train the PS-EOC-TS-MADDPG method, it is necessary to first train the TS-MADDPG method for each trading pair. Then, the trained TS-MADDPG methods are used to train the PS-EOC method. The detail settings of the environment for PS-EOC are given below.

State: We use the change of spread as state for trading pair, and the state for PS-EOC is the historical spread changes of all trading pairs:

$$\boldsymbol{s}_t^{PS-EOC} = \left[\frac{spread_{t-l+1}^1}{spread_{t-l}^1}, \ldots, \frac{spread_t^1}{spread_{t-1}^1}, \ldots, \frac{spread_{t-l+1}^\omega}{spread_{t-l}^\omega}, \ldots, \frac{spread_t^\omega}{spread_{t-1}^\omega}, hold\right]$$

where l is the length of historical data for each trading pair, $hold = 0$ when not holding a position and $hold = 1$ when holding a position.

Action: The agent chooses one of the Ω pairs, and for the convenience of representation, in this paper we label the action of the PS method as ω and the action of the TS method is labeled as $\boldsymbol{a}$.

$$\omega = [1, 2, \ldots, \Omega]$$

Reward: Based on previous research Kim, T., and Kim, H. Y. 2019, Lu et al. (2022); Kim et al. (2022), we set the reward r_t is:

$$NR_t^\omega = \left(\frac{p_{1,t^{short}} - p_{1,t^{long}} - c}{p_{1,t^{long}}} + \frac{p_{2,t^{short}} - p_{2,t^{long}} - c}{p_{2,t^{long}}}\right) \tag{12}$$

$$r_t = R_t(s_t, \omega_t, s_{t+1}) = \begin{cases} 1000 \times NR_t^{\omega_t}, if\ close\ or\ stop - loss \\ 0, otherwise \end{cases} \tag{13}$$

3.3. TS-MADDPG

For each trading pair, we use the MADDPG method with three agents $(agent_1, agent_2, agent_3)$ for trading thresholds setting and call the method TS-MADDPG. MADDPG improves upon traditional Actor-Critic method by enabling it to solve multi-agent problems in mixed cooperative or competitive environment. For an N agent problem, the set of all agent policies is represented as $\pi = \{\pi_1, \ldots, \pi_N\}$, parameterized by $\theta = \{\theta_1, \ldots, \theta_N\}$. The gradient of the expected reward $J(\theta_i) = E[R_i]$ of agent i is written as:

$$\nabla_{\theta_i} J(\mu_i) = \mathbb{E}_{x,a\sim D}\left[\nabla_{\theta_i}\mu_i(\boldsymbol{a}_i|\boldsymbol{o}_i)\nabla_{a_i}Q_i^\mu(x, \boldsymbol{a}_1, \ldots, \boldsymbol{a}_N)|_{\boldsymbol{a}_i=\mu_i(\boldsymbol{o}_i)}\right] \tag{14}$$

The experience replay buffer D contains the tuple $(\boldsymbol{x}, \boldsymbol{x}', \boldsymbol{a}_1, \ldots, \boldsymbol{a}_N, r_1, \ldots, r_N)$. $\boldsymbol{o}_i$ is the observation of the agent i, and $\boldsymbol{x} = [\boldsymbol{o}_1, \ldots, \boldsymbol{o}_n]$ is the

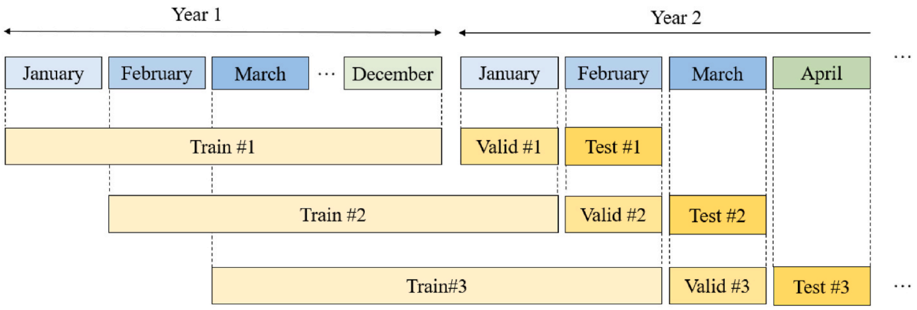


Fig. 4. Example of data division between training set, validation set and test sets.

Table 1
Contracts details.

Contracts	Exchange	Fee	Profit/Point
rb	SHFE	0.00495%	10
p	DCE	2.75CNY	10
MA	CZCE	1.54CNY	10

Table 2
Method name and introduction.

Method type	Method name	Introduction
Static method	Static	Static trading thresholds setting for each pair
	Static-AVG	For all trading pairs, each trading pair using static trading thresholds, averaged
TS method only	TS-DQN	Using DQN for trading thresholds setting for each pair
	TS-DQN-AVG	For all trading pairs, each trading pair using DQN for trading thresholds setting for each pair, averaged
	TS-DDPG	Using DDPG for trading thresholds setting for each pair
	TS-DDPG-AVG	For all trading pairs, each trading pair using DDPG for trading thresholds setting for each pair, averaged
	TS-MADDPG	Using MADDPG for trading thresholds for each pair
	TS-MADDPG-AVG	For all trading pairs, each trading pair using MADDPG for trading thresholds setting for each pair, averaged
PS method only	PS-Coint	Using Cointegration test for pair selection and static trading thresholds for each pair
	PS-DQN	Using DQN for pair selection and static trading thresholds for each pair
	PS-EOC	Using EOC for pair selection and static trading thresholds for each pair
PS-TS method	PS-Coint-TS-DQN	Using Cointegration test for pair selection and DQN for selecting trading thresholds for each pair
	PS-DQN-TS-DQN	Using DQN for pair selection and DQN for selecting trading thresholds for each pair
	PS-EOC-TS-DQN	Using EOC for pair selection and DQN for selecting trading thresholds for each pair
	PS-Coint-TS-MADDPG	Using Cointegration test for pair selection and MADDPG for selecting trading thresholds for each pair
	PS-DQN-TS-MADDPG	Using DQN for pair selection and MADDPG for selecting trading thresholds for each pair
	PS-EOC-TS-MADDPG	Using EOC for pair selection and MADDPG for selecting trading thresholds for each pair

observation vector. $Q_i^{\mu}(\boldsymbol{x}, \boldsymbol{a}_1, \ldots, \boldsymbol{a}_N)$ denotes the centralized state-action value function of agent i, i.e., the critic network, and the centralized action value function Q_i^{μ} is updated as:

$$\mathscr{L}(\theta_i) = \mathbb{E}_{x,a,r,x'}\left[\left(Q_i^{\mu}(\boldsymbol{x}, \boldsymbol{a}_1, \ldots, \boldsymbol{a}_N) - y\right)^2\right], y = r_i + \gamma Q_i^{\mu'}\left(\boldsymbol{x}', \boldsymbol{a}_1', \ldots, \boldsymbol{a}_N'\right)|_{\boldsymbol{a}_j' = \mu_j'(\boldsymbol{o}_j)} \tag{15}$$

where $\mu' = [\mu_1', \ldots, \mu_n']$ is the parameter for which the target strategy has lagged updates θ_j' . It can be seen that the critic network requires global information for learning, whereas the actor uses only local observation information. The above equation uses the policies of other agents, which require constant communication to obtain them, but can also be achieved by estimating the policies of other agents. Each agent maintains $n-1$ policy approximation functions $\widehat{\mu}_{\varphi_i^j}$, which denote agent i to agent j's policy μ_j of the function approximation.

$$\mathscr{L}(\varnothing_i^j) = -\mathbb{E}_{o_j,a_j}\left[log\ \widehat{\mu}_i^j(\boldsymbol{a}_j|\boldsymbol{o}_j) + \lambda H(\widehat{\mu}_i^j)\right] \tag{16}$$

where H is the entropy of the strategy. Replace y in the above equation with $\widehat{y}$:

$$\widehat{y} = r_i + \gamma Q_i^{\mu'}\left(\boldsymbol{x}', \widehat{\mu}_1^{'1}(\boldsymbol{o}_1), \ldots, \mu_i'(\boldsymbol{o}_i), \ldots, \widehat{\mu}_1^{'N}(\boldsymbol{o}_N)\right) \tag{17}$$

Before Q_i^{μ} update, we fetch each agent j from the buffer to perform a single gradient step to update the latest sample of $\widehat{\mu}_{\varphi_i^j}$.

In the TS-MADDPG method, $agent_1$ is responsible for setting the opening threshold, while $agent_2$ is in charge of determining the stop-loss threshold. $agent_3$ is responsible for evaluating whether the thresholds set by $agent_1$ and $agent_2$ are valid at the current time step t. Through cooperation, the three agents work together to maximize the total return. For the TS-MADDPG strategy corresponding to the trading pair ω, the environment is configured as follows.

State: All agents' observations are the historical price of the trading pair:

$$\boldsymbol{s}_t^{TS-MADDPG} = \left[\frac{spread_{t-l+1}^{\omega}}{spread_{t-l}^{\omega}}, \ldots, \frac{spread_t^{\omega}}{spread_{t-1}^{\omega}}, hold\right]$$

Action: $agent_1$ and $agent_2$ need to choose one of the given open and stop-loss thresholds. $agent_3$ need to decide whether to trade at the current time step, when $agent_3$ output 1, the pair trading strategy is executed, when $agent_3$ output 0, all trading operations are blocked.

$$a^{agent_1} = [0.5, 1.0, 1.5, 2.0, 2.5, 3.0]$$

$$a^{agent_2} = [1.5, 2.0, 2.5, 3.0, 3.5, 4.0]$$

$$a^{agent_3} = [0, 1]$$

$$\boldsymbol{a}_t^{\omega_t} = [a^{agent_1}, a^{agent_2}, a^{agent_3}]$$

Reward: the reward functions of all three agents are the same as those of the PS-EOC.

4. Simulations

4.1. Data

For the simulation, 5-min level data from the Chinese futures market was utilized. Considering the correlation between futures and different

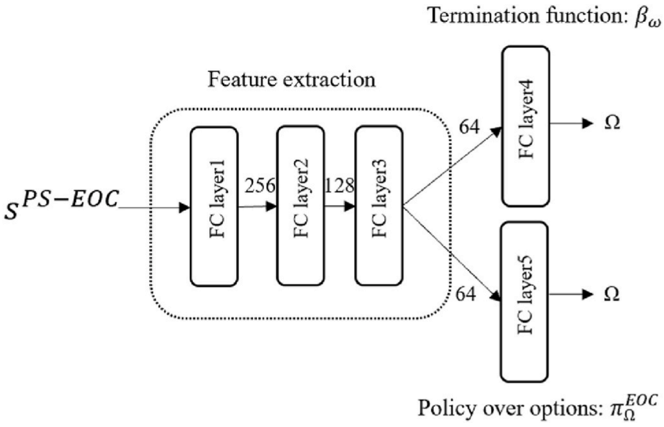


Fig. 5. The network architecture of the EOC using in simulations.

futures exchanges, three commodity futures (rb, p, and MA) were selected as the candidate trading pairs. Trading was carried out on a monthly basis, and the spread of each pair was comprised of the two most active contracts for that commodity in each month. The dataset covers a period of 37 months, from January 2020 to January 2023.

In terms of training, to capture the cyclical behavior exhibited by futures commodities, we employed a rolling forward approach. A one-month period was used as the test set, with the previous month's data serving as the validation set, and the data from the previous year used as the training set. This method ensured that a separate model was trained for each month's data in the test set. Train multiple times on the training set and select the best performance on the validation set for testing. From February 2021 to January 2023, there are a total of 24 test months were used. Fig. 4 provides an example illustrating the division of data between the training set, validation set, and test set.

Details of each contract are given in Table 1. For the commission type, a fixed commission is charged when it is of a fixed type. For instance, when dealing with p contract, a fee of 2.75CNY is charged, i.e., transaction cost $c = 2.75$. On the other hand, if it is of the proportional type, a fixed percentage of the transaction amount is charged as commission. For example, when rb contract is opened at 3500, the fee is calculated as 3500 multiplied by 10 and then multiplied by 0.00495%, and one can obtain transaction cost $c = 1.73$. To calculate indicators, we set the starting capital to 20000 CNY (US $ 2900).

4.2. Model setting

PyTorch was used to train and test the models on a server equipped with an Intel Xeon Silver 4214R, NVIDIA RTX 3090, and 256 GB of RAM. In addition to the previously mentioned PS-EOC and TS-MADDPG models, we included PS-DQN, TS-DQN, TS-DDPG, PS-Coint, and Static methods for comparison. Table 2 summarizes all methods and their introductions.

In Table 2, TS-DQN is based on Kim, T., and Kim, H. Y. (2019), which we modified to facilitate comparison with other models. The AVG method was used for comparison with the PS method, and it is important to note that the PS method does not allow opening multiple positions simultaneously, while the AVG method is equivalent to allowing multiple positions to be opened simultaneously. The detailed settings for all models are listed below.

Static: The opening and stop-loss thresholds for the static strategy are set to 0.5 and 4.0, trading window W set to 100, respectively, which we have obtained the highest returns for the three pairs corresponding to the static parameter strategies. For the static parameters, smaller opening thresholds and larger stop loss thresholds usually lead to higher returns, which are brought about by a higher number of trades.

TS-DQN & TS-MADDPG: The settings for TS-DQN and TS-MADDPG were the same, with both the replay buffer size set to 20000, the batch size to 256, the GAMMA to 0.995, and the learning rate to 0.0001 (all networks of MADDPG had the same learning rate), using the Adam optimizer and 50 episodes of training.

TS-DDPG: Replay buffer size set to 20000, the batch size to 256, the GAMMA to 0.995, and the learning rate for the actor network and critic network are set to 0.0001. Using the Adam optimizer and 50 episodes of training. The output of TS-DDPG are two continuous values between 0 and 1. We scale it to the following range for comparison with other models:

$$ol \in [0.5, 3.0], sl \in [1.5, 4.0]$$

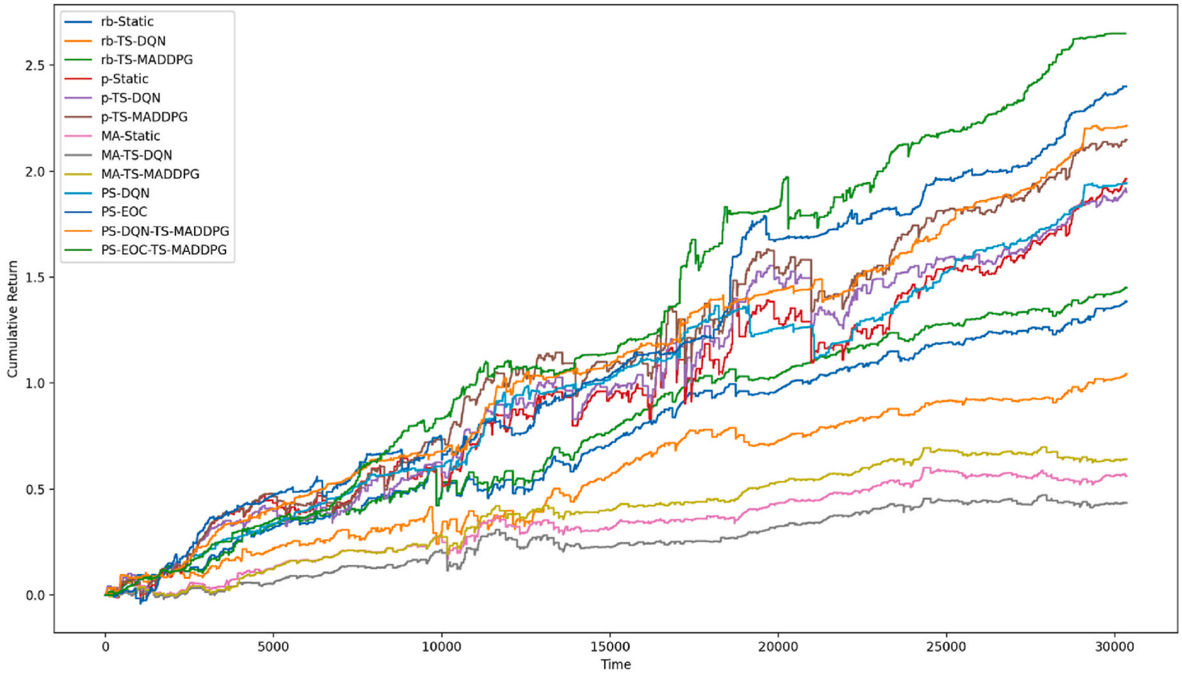


Fig. 6. CR of main methods.

Table 3
TS method results.

Pairs	Method	TN	WN	WR	SR	MDD	P/L	CR
rb	Static	1265	1210	0.957	1.873	**−0.106**	0.750	1.386
	TS-DQN	1021	954	0.934	1.817	−0.124	0.694	1.047
	TS-DDPG	292	278	0.955	2.441	−0.125	0.663	0.361
	TS-MADDPG	1161	1128	0.972	1.829	**−0.106**	**0.757**	**1.450**
p	Static	1001	938	0.937	**1.848**	−0.123	0.671	1.964
	TS-DQN	896	802	0.895	1.820	−0.117	0.625	1.902
	TS-DDPG	406	383	0.943	2.208	−0.236	0.606	0.701
	TS-MADDPG	918	887	0.966	1.817	**−0.112**	**0.689**	**2.150**
MA	Static	1124	1054	0.938	**1.862**	−0.088	0.678	0.562
	TS-DQN	882	808	0.916	1.683	−0.081	0.639	0.435
	TS-DDPG	357	325	0.908	1.561	−0.080	0.525	0.156
	TS-MADDPG	967	934	0.965	1.752	**−0.072**	**0.685**	**0.643**
Static-AVG		1130	1067	0.944	1.861	−0.123	0.700	1.304
TS-DQN-AVG		933	854	0.915	1.773	−0.124	0.653	1.128
TS-DDPG-AVG		352	329	0.935	2.070	−0.147	0.598	0.406
TS-MADDPG-AVG		1015	983	0.968	1.799	−0.112	0.710	1.414

Table 4
PS method results.

Method	TN	WN	WR	SR	MDD	P/L	CR
p-Static	1001	938	0.937	1.848	−0.123	0.671	1.964
p-TS-DQN	896	802	0.895	1.82	−0.117	0.625	1.902
p-TS-MADDPG	918	887	0.966	1.817	−0.112	0.689	2.150
Static-AVG	1130	1067	0.944	**1.861**	−0.123	0.700	1.304
TS-DQN-AVG	933	854	0.915	1.773	−0.124	0.653	1.128
TS-MADDPG-AVG	1015	983	0.968	1.799	−0.112	0.710	1.414
PS-Coint	1323	1138	0.860	1.649	−0.112	**0.732**	1.408
PS-DQN	1358	1030	0.758	1.814	−0.107	0.663	1.952
PS-EOC	1272	1140	0.896	1.683	**−0.071**	0.700	**2.400**
PS-EOC (in-sample)	1314	1192	0.907	1.663	−0.091	0.718	2.776

Table 5
PS-TS method results.

Method	TN	WN	WR	SR	MDD	P/L	CR
p-Static	1001	938	0.937	1.848	−0.123	0.671	1.964
p-TS-DQN	896	802	0.895	1.82	−0.117	0.625	1.902
p-TS-MADDPG	918	887	0.966	1.817	−0.112	0.689	2.150
Static-AVG	1130	1067	0.944	**1.861**	−0.123	0.700	1.304
TS-DQN-AVG	933	854	0.915	1.773	−0.124	0.653	1.128
TS-MADDPG-AVG	1015	983	0.968	1.799	−0.112	0.710	1.414
PS-Coint-TS-DQN	987	915	0.927	1.628	−0.115	0.721	1.321
PS-DQN-TS-DQN	956	796	0.833	1.800	−0.087	0.674	1.876
PS-EOC-TS-DQN	873	791	0.906	1.856	−0.094	0.673	2.288
PS-Coint-TS-MADDPG	1057	1025	0.970	1.600	−0.086	0.731	1.434
PS-DQN-TS-MADDPG	1371	1047	0.764	1.789	**−0.044**	0.669	2.224
PS-EOC-TS-MADDPG	1051	970	0.920	1.617	−0.082	**0.729**	**2.650**
PS-EOC-TS-MADDPG (in-sample)	1010	924	0.915	1.813	−0.081	0.714	2.810

PS-Coint: Using cointegration test to select trades, for each month in the test set, we perform the cointegration test using data from the previous month and select the trading pair with the highest confidence level.

PS-DQN: The state of PS-DQN is different from PS-EOC, as PS-DQN selects trading pairs every fixed time interval. Below are the settings of the PS-DQN environment.

State: The state of PS-DQN is the change in spread during the trading period $(t - L, t)$.

$$s_t^{PS-DQN} = \left[\frac{spread^1_{t-L}}{spread^1_{t-L-1}}, \dots, \frac{spread^1_t}{spread^1_{t-1}}, \dots, \frac{spread^\omega_{t-L}}{spread^\omega_{t-L-1}}, \dots, \frac{spread^\omega_t}{spread^\omega_{t-1}}, hold\right]$$

Action: Same as PS-EOC.

Reward: The reward of PS-DQN is the sum of all rewards during the trading period $(t, t + L)$.

$$r_t^{PS-DQN} = R_t^{PS-DQN}\left(s_t^{PS-DQN}, \omega_t, s_{t+L}^{PS-DQN}\right) = \sum\nolimits_t^{t+L} r_t^{PS-EOC}$$

In this paper, we set the fixed time interval L to 100. The replay buffer size is 6400, the batch size is 64, the GAMMA is 0.995, and using Adam optimizer, the learning rate of 0.0001, and 50 episodes of training each time. The network architecture of PS-DQN consists of four fully connected layers with output sizes of 512, 256, 128, and Ω, respectively.

PS-EOC: The replay buffer size is set to 20000, the batch size is 256, the GAMMA is 0.995, the learning rate is 0.0001, and the RMSprop optimizer is used to train 50 episodes each time. PS-EOC also has three special hyperparameters *minsteps*, η and α, where *minsteps* is set to 100 to facilitate comparison with other models. η and α are set to 0.2 and 0.3, respectively. We will discuss the last two parameters in Section 4.5.3.

Fig. 5 illustrates the network architecture of the EOC model, where the output sizes of the fully connected layers are 256, 128, 64, Ω, Ω, respectively.

When both methods are used, each method has the same settings as the single method described above. We make the following assumptions for our simulation: all trades are executed instantly without any slippage, and our trades do not impact the market.

4.3. Evaluation indicators

The performance of our proposed approach is evaluated by a set of indicators including Number of trades, Number of successes, Win Rate, Sharpe ratio, Maximum drawdown, Profit-loss ratio, and Cumulative return.

Number of trades (TN): The number of times a position is opened by hitting the opening threshold.

Number of successes (WN): The number of times the position was closed by mean reversion. It should be noted that, when mean reversion occurs, there is no guarantee of a positive profit.

Table 6
Monthly trading results for PS-EOC-TS-MADDPG.

Month	TN	WN	WR	SR	MDD	P/L	CR
Feb-21	35	28	0.800	1.179	−0.003	0.722	0.044
Mar-21	53	47	0.887	2.255	−0.015	0.724	0.071
Apr-21	45	35	0.778	1.951	−0.003	0.612	0.132
May-21	50	49	0.980	1.942	−0.007	0.849	0.088
Jun-21	59	58	0.983	2.319	−0.022	0.758	0.031
Jul-21	65	63	0.969	1.452	−0.012	0.814	0.169
Aug-21	42	36	0.857	2.058	−0.017	0.674	0.186
Sep-21	25	25	1.000	1.779	−0.047	0.815	0.088
Oct-21	42	41	0.976	1.941	−0.034	0.783	0.207
Nov-21	38	37	0.974	2.605	−0.067	0.609	0.051
Dec-21	38	36	0.947	−0.252	−0.045	0.587	0.006
Jan-22	39	39	1.000	4.549	−0.004	0.878	0.076
Feb-22	29	23	0.793	1.925	−0.007	0.645	0.058
Mar-22	75	66	0.880	1.138	−0.042	0.788	0.414
Apr-22	32	31	0.969	0.541	−0.138	0.722	0.195
May-22	54	53	0.981	−0.558	−0.019	0.797	0.022
Jun-22	29	29	1.000	0.473	−0.215	0.781	−0.042
Jul-22	38	36	0.947	0.967	−0.070	0.682	0.135
Aug-22	56	53	0.946	1.399	−0.048	0.742	0.188
Sep-22	40	36	0.900	2.457	−0.017	0.667	0.068
Oct-22	24	23	0.958	1.088	−0.011	0.643	0.056
Nov-22	55	50	0.909	1.682	−0.012	0.793	0.177
Dec-22	53	41	0.774	1.553	−0.001	0.788	0.201
Jan-23	38	35	0.921	1.747	−0.005	0.846	0.029

Win Rate (WR) : $WR = \dfrac{WN}{TN}$

Sharpe ratio (SR): An indicator that reflects risk-adjusted returns.

$$SharpeRation = \frac{E(R_p) - R_f}{\sigma_p}$$

Where $E(R_p)$ is the annualized rate of return, and R_f is the annualized risk-free rate, and σ_p is the standard deviation of the annualized rate of return.

Maximum drawdown (MDD): The maximum loss from a peak to a trough before reaching a new peak.

Profit-loss ratio (P/L): The ratio of the average amount of profit to the average amount of loss.

Cumulative return (CR): The sum of the returns on all test sets.

4.4. Simulation results

Fig. 6 presents the CR of the main methods. Our proposed PS-EOC-TS-MADDPG method achieves the highest CR. Next, we compare the effectiveness of each model from various aspects.

To begin with, a comparison is made between the results of TS methods, where the simulation results of Static, TS-DQN, TS-DDPG and TS-MADDPG are presented in Table 3.

The simulation results in Table 3 indicate that the TS-MADDPG method outperforms the Static and TS-DQN methods in terms of WR and P/L ratios, and also shows a reduced MDD. On average, the TS-MADDPG method achieves a 10% improvement in CR while maintaining a lower number of trades compared to the Static method. In contrast, the TS-DQN method generally exhibits a lower number of trades and relatively lower returns. For the TS-DDPG method, we observed that the agent tends to always choose the maximum opening and stop-loss thresholds, resulting in too few trades and making it difficult to gain profits. In the subsequent comparisons, we no longer use the TS-DDPG method.

Secondly, the comparison of PS methods is presented in Table 4.

The simulation results presented in Table 4 demonstrate that the PS-EOC method achieved a 19% improvement in returns compared to PS-DQN. Compared to the p-Static method with the highest CR in the Static methods, it achieves an improvement of 22%. Due to the inability to select trading pairs during trading, the PS-Coint method only shows an improvement in CR compared to the Static-AVG method. Combining the results of the PS-DQN and PS-EOC methods, it can be concluded that dynamically selecting trading pairs can improve returns.

Finally, the results of PS-TS methods are compared in Table 5.

The results presented in Table 5 demonstrate that the PS-EOC-TS-MADDPG method achieves a remarkable performance, yielding 2.03 times CR than the Static-AVG with a smaller number of trades. When compared to the p-Static method with the highest CR in Static methods, there is a 29% improvement in CR. Compared to the p-TS-MADDPG method with the highest CR in the TS methods, there is a 19% improvement in CR. Moreover, when compared to the PS-DQN-TS-MADDPG method, the PS-EOC-TS-MADDPG method exhibits a higher

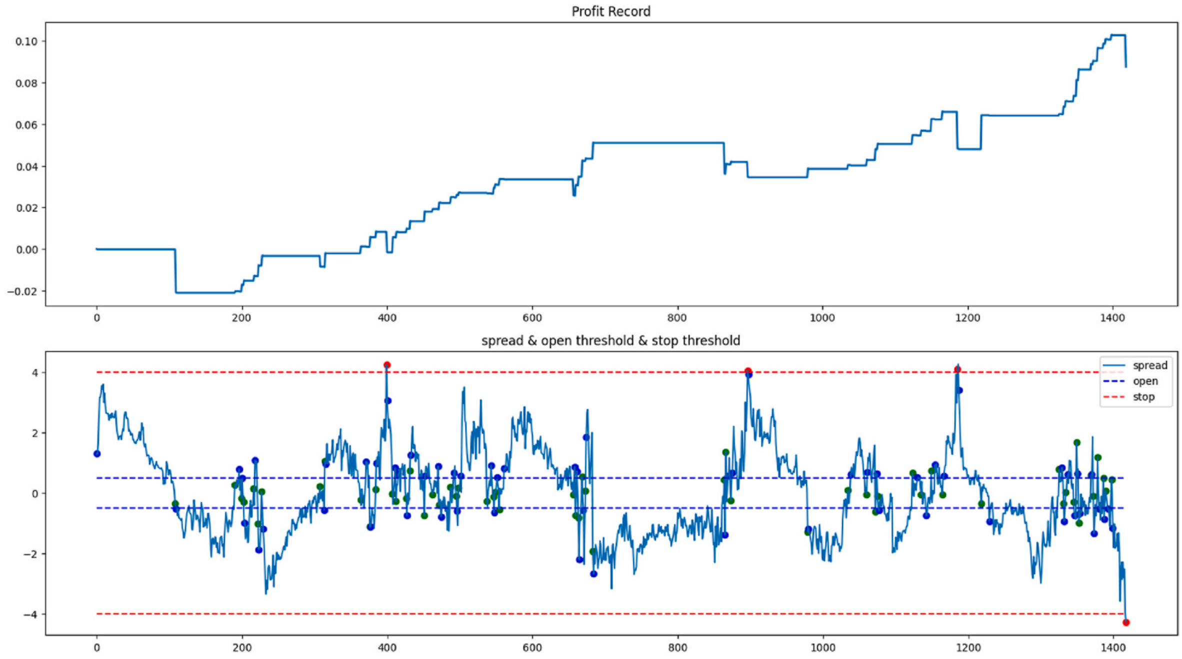


Fig. 7. rb-Static trading record in December 2022.

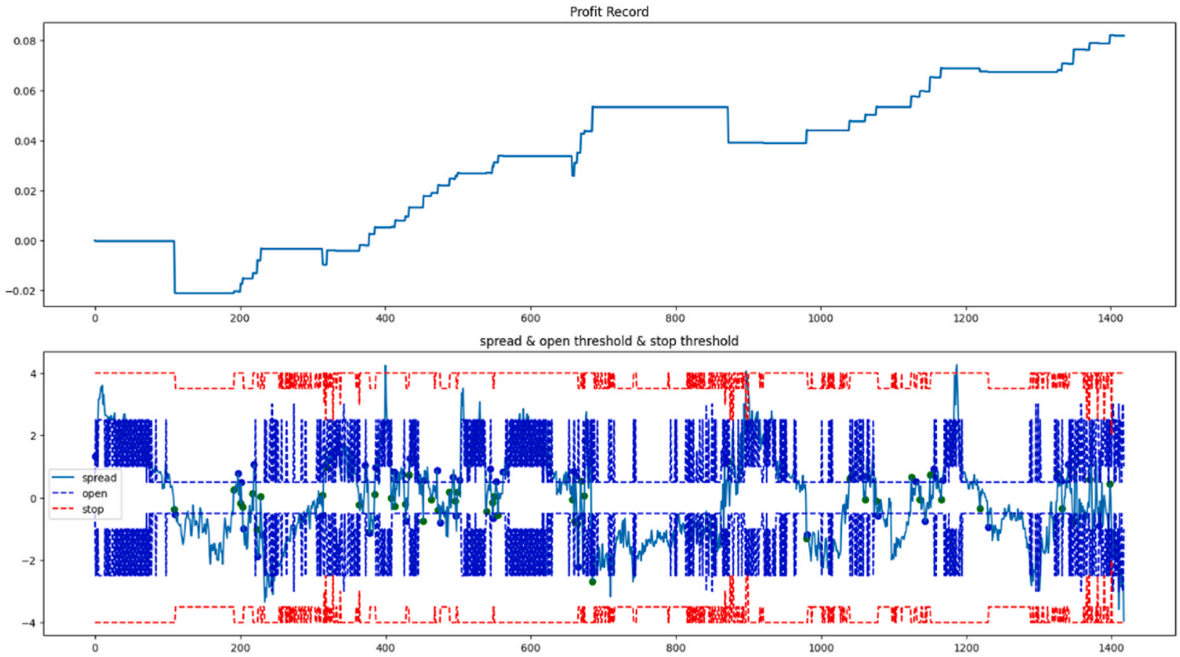


Fig. 8. rb-TS-MADDPG trading record in December 2022.

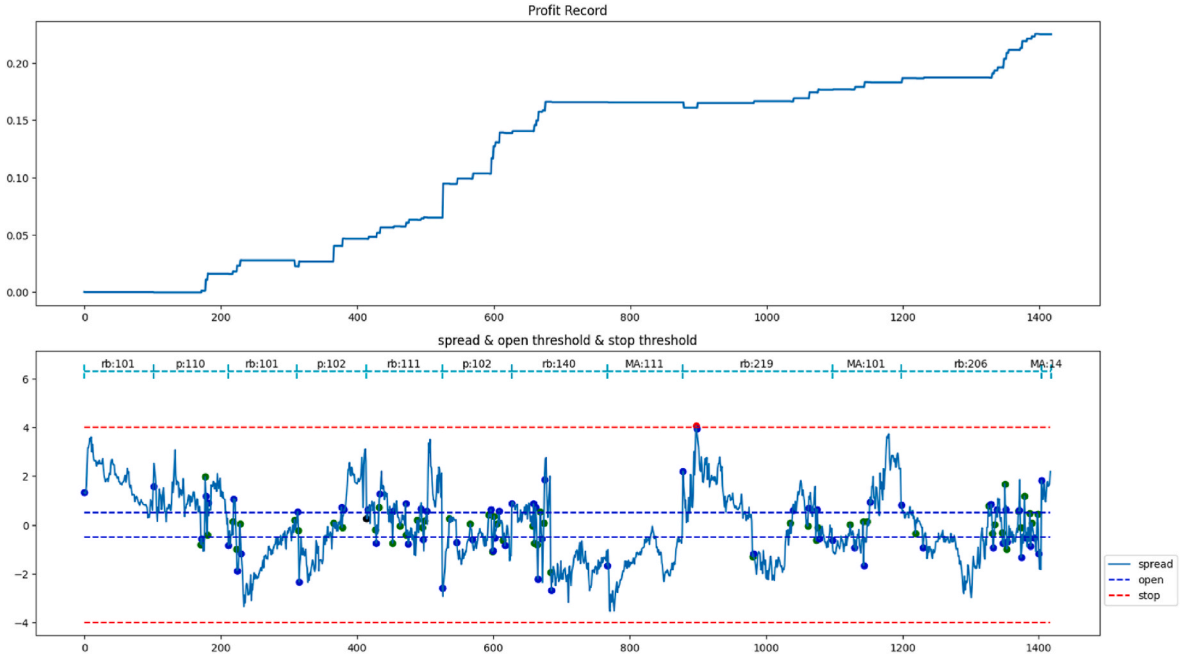


Fig. 9. PS-EOC trading record in December 2022.

P/L ratio and a 16% improvement in CR. When the TS method is changed to TS-DQN, PS-EOC-TS-DQN shows a 20% improvement compared to PS-DQN, demonstrating the effectiveness of PS-EOC in selecting trading pairs over PS-DQN. And, monthly trading results for PS-EOC-TS-MADDPG from Feb-21 to Jan-23 are shown in Table 6. The indicators in Table 6 are calculated with a monthly starting capital of 20000 CNY.

4.5. Model analyze

4.5.1. Trading records visualization

In this section, we present the effectiveness of our model using the trading records visualization on the test set of December 2022 as an example.

Fig. 7 illustrates the trading record of trading pair rb using static thresholds in December 2022. In Fig. 7 and the following Figs. 8–10, the upper part shows the changes in profits during the month's trading. The lower part depicts the green line representing the spread, the blue line representing the opening threshold, and the red line representing the stop-loss threshold. Blue points represent opening points, green points represent closing positions through mean reversion, and red points represent stop-loss points by hitting stop-loss thresholds.

Fig. 8 provides an example of trading thresholds for rb using the TS-MADDPG method on the test set in December 2022. Compared with

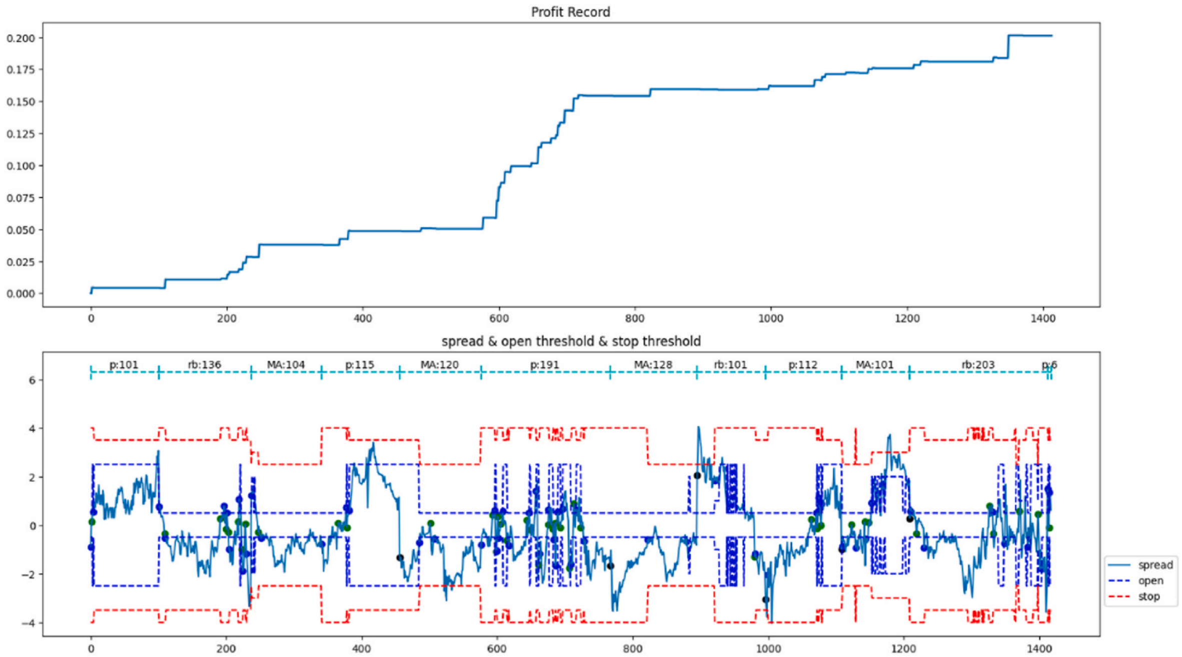


Fig. 10. PS-EOC-TS-MADDPG trading record in December 2022.

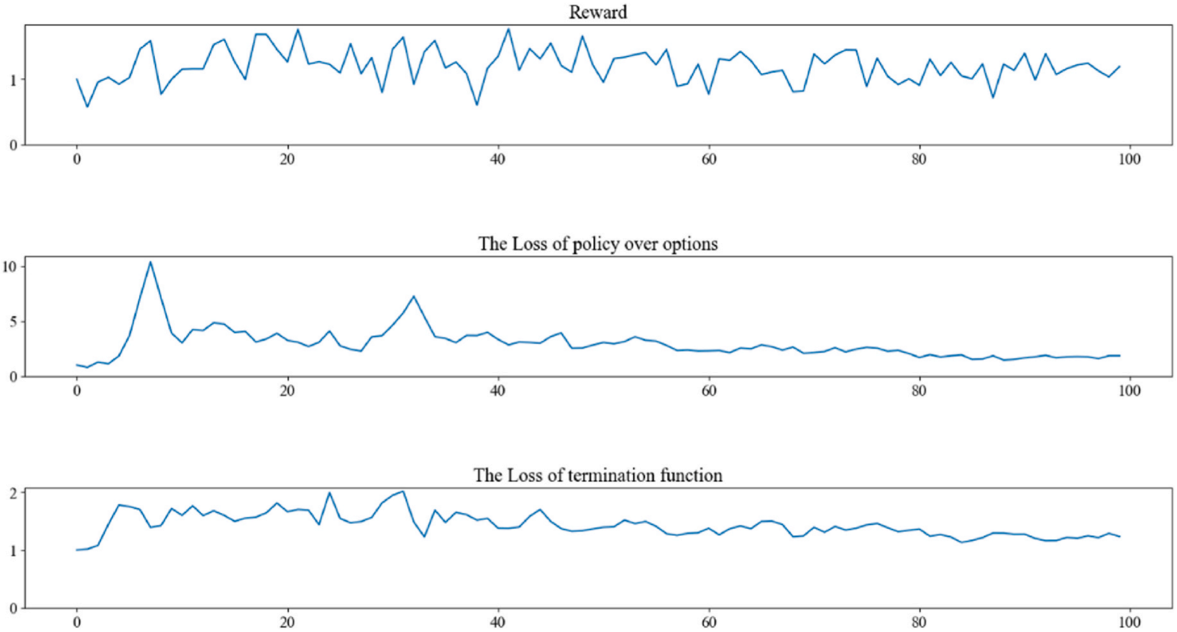


Fig. 11. The changes in total rewards, the loss of policy over options loss, and loss of termination function for 100 episodes.

Fig. 7, it can be observed that the agent automatically adjusts the threshold according to spread fluctuations to improve returns. Notably, the stop-loss threshold output by the agent is higher than or equal to the opening threshold, which is obtained by the agent through learning, even if it is not set on the reward function.

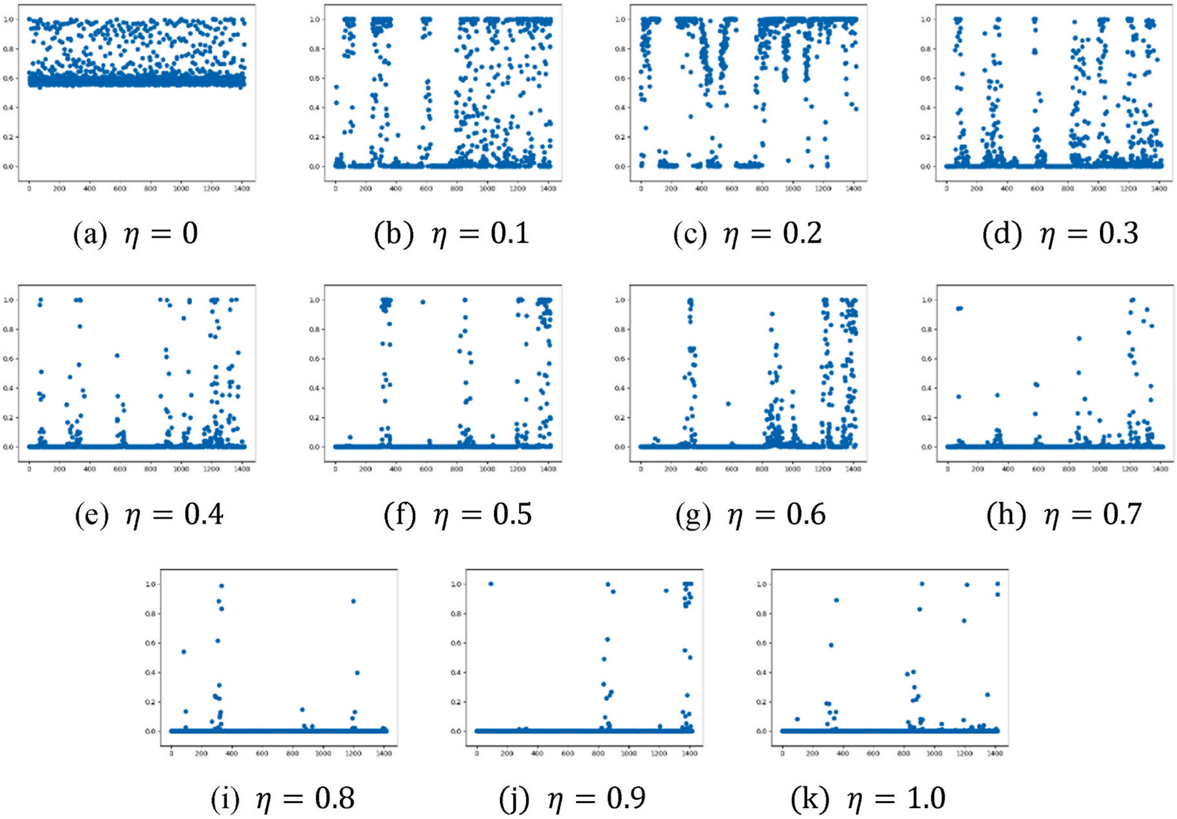
Fig. 9 presents the trading record of PS-EOC in December 2022. In the lower part of Fig. 9, the line above represents which trading pair was selected by EOC and the duration of the trade in time steps. This demonstrates that EOC can automatically determine when to terminate the current trading pair and select profitable trading pairs. Compared with Fig. 7, when rb-Static performs poorly in the middle of the month and may not yield favorable results if trading at that time, the PS-EOC agent selects to trade MA and p, which effectively improves the returns.

Fig. 10 presents the trading record of TS-EOC-TS-MADDPG on test set in December 2022. By comparing with Fig. 9, it can be observed that the PS-EOC-TS-MADDPG method not only enables more flexible pair selection but also enables timely adjustment of trading thresholds.

4.5.2. *Convergence test*

The following Fig. 11 illustrates the changes in total rewards, the loss of policy over options, and the loss of termination function during continuous training for 100 episodes when the test set is for December 2022.

In Fig. 11, we assume that the first point starts from 1. Due to the exploration conducted by the agent throughout the entire training process using Bernoulli distribution sampling and ϵ-greedy policy, there



(a) $\eta = 0$ (b) $\eta = 0.1$ (c) $\eta = 0.2$ (d) $\eta = 0.3$

(e) $\eta = 0.4$ (f) $\eta = 0.5$ (g) $\eta = 0.6$ (h) $\eta = 0.7$

(i) $\eta = 0.8$ (j) $\eta = 0.9$ (k) $\eta = 1.0$

Fig. 12. The effect of the value of η on the termination probability.

is fluctuation in the rewards on the training set in each episode. However, after 50 episodes, the loss of policy over options and the loss of termination function stabilizes, indicating that the agent's policy has reached a relatively stable state.

4.5.3. *Parameter discussion*

In this subsection, we discuss two hyperparameters of the EOC method: η and α, using the test results of the PS-EOC method on December 2022. η is the correction term in the loss function of the termination function model when training the EOC model. α is the lower confidence limit of the termination probability output by the termination function.

The following Fig. 12 gives the termination probability of the termination function output at each time point when $\eta \in [0, 1]$ with a step size of 0.1.

Fig. 12 evident that the termination probability of the output of the termination function gradually approaches 0 as the value of η increases. Our simulation shows that setting a small value for η (e.g., $\eta = 0$) results in the agent terminating the current trading pair at every time point, while a larger value for η (e.g., $\eta = 1.0$) results in the agent maintaining the current trading pair. The choice of η depends on our preference for using the model: if we prefer to switch trading pairs frequently, we can set η to a smaller value, and a smaller α can be chosen to further increase the switching frequency of trading pairs. On the other hand, if we prefer to maintain the current trading pair, we can set η to a larger value and a larger α can be chosen to reduce the switching frequency further. It is advisable to adjust η and α based on the validation set to achieve the highest return in actual trading. In this paper, we have chosen $\eta = 0.2$ and $\alpha = 0.3$ after testing against all validation sets to standardize the model effects and enable comparisons.

5. Conclusions

This paper presents a two-level framework for improving pair trading strategies by utilizing the EOC for pair selection and MADDPG method for trade thresholds setting. The effectiveness of our proposed approach is demonstrated through multiple simulation in the Chinese futures market.

The issue of varying returns for the same trading pairs at different time periods in pairs trading strategies is addressed. Our simulations show that using DQN to select pairs at a fixed time interval results in a 50% improvement in average CR compared to Static-AVG method. However, the duration of better performance is not fixed. To overcome this limitation, we introduce an EOC approach that allows the agent to learn the termination function and determine when to close the current pair. This approach leads to a further 23% increase in CR compared to the DQN method.

Additionally, we employ the MADDPG method to select thresholds based on previous studies, which takes advantage of communication cooperation among multiple agents to enhance gains. This approach outperforms the DQN method with a 5%–15% improvement in CR.

Finally, we combine both methods to improve the pairs trading strategy from both sides simultaneously. When compared with the highest CR in Static methods, there is a 29% improvement in CR, and a 16% improvement compared to the DQN method.

Two areas for future improvement are proposed. First, providing a more stable representation of the state by utilizing better feature selection or extraction methods to better understand the complex market environment. Second, we suggest that our proposed EOC approach can be applied to a wider range of strategy selection problems, although further validation simulations are necessary.

CRediT authorship contribution statement

Zhizhao Xu: Conceptualization, Software, Data curation, Writing – original draft. **Chao Luo:** Methodology, Writing – review & editing, Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgements

This research is supported by the National Natural Science Foundation of China (Nos: 62172264); Shandong Provincial Natural Science Foundation (ZR2019MF020).

References

Bacon, P.L., Harb, J., Precup, D., 2017. The option-critic architecture. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 31, 1.
Bertram, W.K., 2010. Analytic solutions for optimal statistical arbitrage trading. Phys. Stat. Mech. Appl. 389 (11), 2234–2243.
Brim, A., 2020. Deep reinforcement learning pairs trading with a double deep Q-network. In: 2020 10th Annual Computing and Communication Workshop and Conference (CCWC). IEEE, pp. 222–227.
Chen, H., Chen, S., Chen, Z., Li, F., 2019. Empirical investigation of an equity pairs trading strategy. Manag. Sci. 65 (1), 370–389.
Deng, Y., Bao, F., Kong, Y., Ren, Z., Dai, Q., 2016. Deep direct reinforcement learning for financial signal representation and trading. IEEE Transact. Neural Networks Learn. Syst. 28 (3), 653–664.
Do, B., Faff, R., 2012. Are pairs trading profits robust to trading costs? J. Financ. Res. 35 (2), 261–287.
Elliott, R.J., Van Der Hoek*, J., Malcolm, W.P., 2005. Pairs trading. Quant. Finance 5 (3), 271–276.
Fallahpour, S., Hakimian, H., Taheri, K., Ramezanifar, E., 2016. Pairs trading strategy optimization using the reinforcement learning method: a cointegration approach. Soft Comput. 20, 5051–5066.
Fujimoto, S., Hoof, H., Meger, D., 2018. Addressing function approximation error in actor-critic methods. In: International Conference on Machine Learning. PMLR, pp. 1587–1596.
Galenko, A., Popova, E., Popova, I., 2012. Trading in the presence of cointegration. J. Altern. Investments 15 (1), 85–97.
Gatev, E., Goetzmann, W.N., Rouwenhorst, K.G., 2006. Pairs trading: performance of a relative-value arbitrage rule. Rev. Financ. Stud. 19 (3), 797–827.
Hedrea, E.L., Precup, R.E., Roman, R.C., Petriu, E.M., 2021. Tensor product-based model transformation approach to tower crane systems modeling. Asian J. Control 23 (3), 1313–1323.
Hochreiter, S., Schmidhuber, J., 1997. Long short-term memory. Neural Comput. 9 (8), 1735–1780.
Kim, T., Kim, H.Y., 2019. Optimizing the pairs-trading strategy using deep reinforcement learning with trading and stop-loss boundaries. Complexity 2019, 1–20.
Kim, S.H., Park, D.Y., Lee, K.H., 2022. Hybrid deep reinforcement learning for pairs trading. Appl. Sci. 12 (3), 944.
Leung, T., Li, X., 2013. Optimal mean reversion trading with transaction costs and stop-loss exit. Int. J. Theor. Appl. Finance 18 (3).
Li, X., Cui, C., Cao, D., Du, J., Zhang, C., 2022. Hypergraph-based reinforcement learning for stock portfolio selection. In: ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, pp. 4028–4032.
Lin, Y.X., McCrae, M., Gulati, C., 2006. Loss protection in pairs trading through minimum profit bounds: a cointegration approach. Advances in Decision Sciences 2006.
Lillicrap, T.P., Hunt, J.J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., Wierstra, D., 2016. Continuous control with deep reinforcement learning. In: Bengio, Y., LeCun, Y. (Eds.), 4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings.
Lin, Y.C., Chen, C.T., Sang, C.Y., Huang, S.H., 2022. Multiagent-based deep reinforcement learning for risk-shifting portfolio management. Appl. Soft Comput. 123, 108894.
Lowe, R., Wu, Y.I., Tamar, A., Harb, J., Pieter Abbeel, O., Mordatch, I., 2017. Multi-agent actor-critic for mixed cooperative-competitive environments. Adv. Neural Inf. Process. Syst. 30.
Lu, J.Y., Lai, H.C., Shih, W.Y., Chen, Y.F., Huang, S.H., Chang, H.H., et al., 2022. Structural break-aware pairs trading strategy using deep reinforcement learning. J. Supercomput. 78 (3), 3843–3882.
Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., Riedmiller, M., 2013. Playing atari with deep reinforcement learning. In: Proc. Adv. Neural Information Processing Systems Workshop on Deep Learning.
Panagopoulos, O.P., Xanthopoulos, P., Razzaghi, T., Şeref, O., 2019. Relaxed support vector regression. Ann. Oper. Res. 276, 191–210.
Pole, A., 2011. Statistical Arbitrage: Algorithmic Trading Insights and Techniques. John Wiley & Sons.
Pozna, C., Precup, R.E., 2012. Aspects concerning the observation process modelling in the framework of cognition processes. Acta Polytechnica Hungarica 9 (1), 203–223.
Pozna, C., Minculete, N., Precup, R.E., Kóczy, L.T., Ballagi, Á., 2012. Signatures: definitions, operators and applications to fuzzy modelling. Fuzzy Set Syst. 201, 86–104.
Precup, D., 2000. Temporal Abstraction in Reinforcement Learning. University of Massachusetts, Amherst.
Precup, R.E., Duca, G., Travin, S., Zinicovscaia, I., 2022. Processing, neural network-based modeling of biomonitoring studies data and validation on Republic of Moldova data. Proc. Rom. Acad. Math. Phys. Tech. Sci. Inf. Sci. 23 (4), 403–410.
Puterman, M.L., 2014. Markov Decision Processes: Discrete Stochastic Dynamic Programming. John Wiley & Sons. Stolle, M., & Precup, D. (2002). Learning options in reinforcement learning. In Abstraction, Reformulation, and Approximation: 5th International Symposium, SARA 2002 Kananaskis, Alberta, Canada August 2–4, 2002 Proceedings 5 (pp. 212–223). Springer Berlin Heidelberg.
Sarmento, S.M., Horta, N., 2020. Enhancing a pairs trading strategy with the application of machine learning. Expert Syst. Appl. 158, 113490.
Şeref, O., Razzaghi, T., Xanthopoulos, P., 2017. Weighted relaxed support vector machines. Ann. Oper. Res. 249, 235–271.
Shavandi, A., Khedmati, M., 2022. A multi-agent deep reinforcement learning framework for algorithmic trading in financial markets. Expert Syst. Appl. 208, 118124.
Sutton, R.S., Precup, D., Singh, S., 1999. Between MDPs and semi-MDPs: a framework for temporal abstraction in reinforcement learning. Artif. Intell. 112 (1–2), 181–211.
Van Hasselt, H., Guez, A., Silver, D., 2016. Deep reinforcement learning with double q-learning. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 30, 1.
Vidyamurthy, G., 2004. Pairs Trading: Quantitative Methods and Analysis, vol. 217. John Wiley & Sons.
Wang, J., Zhang, Y., Tang, K., Wu, J., Xiong, Z., 2019. Alphastock: a buying-winners-and-selling-losers investment strategy using interpretable deep reinforcement attention networks. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 1900–1908.
Wang, C., Sandås, P., Beling, P., 2021. Improving pairs trading strategies via reinforcement learning. In: 2021 International Conference on Applied Artificial Intelligence (ICAPAI). IEEE, pp. 1–7.
Winkel, D., Strauß, N., Schubert, M., Seidl, T., 2022. Risk-aware reinforcement learning for multi-period portfolio selection. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases. Springer Nature Switzerland, Cham, pp. 185–200.
Zha, L., Dai, L., Xu, T., Wu, D., 2022. A hierarchical reinforcement learning framework for stock selection and portfolio. In: 2022 International Joint Conference on Neural Networks (IJCNN). IEEE, pp. 1–7.