Expert Systems With Applications 221 (2023) 119707

Contents lists available at ScienceDirect

Expert Systems With Applications

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

Asset correlation based deep reinforcement learning for the portfolio selection

Tianlong Zhao [a], Xiang Ma [a], Xuemei Li [a], Caiming Zhang [a,b,c,*]

[a] School of Software, Shandong University, Jinan 250101, China
[b] Shandong Co-Innovation Center of Future Intelligent Computing, Yantai 264025, China
[c] Digital Media Technology Key Lab of Shandong Province, Jinan 250014, China

ARTICLE INFO

Keywords:
Portfolio selection
Deep reinforcement learning
Asset correlations

ABSTRACT

Portfolio selection is an important application of AI in the financial field, which has attracted considerable attention from academia and industry alike. One of the great challenges in this application is modeling the correlation among assets in the portfolio. However, current studies cannot deal well with this challenge because it is difficult to analyze complex nonlinearity in the correlation. This paper proposes a policy network that models the nonlinear correlation by utilizing the self-attention mechanism to better tackle this issue. In addition, a deterministic policy gradient recurrent reinforcement learning method based on Monte Carlo sampling is constructed with the objective function of cumulative return to train the policy network. In most existing reinforcement learning-based studies, the state transition probability is generally regarded as unknown, so the value function of the policy can only be estimated. Based on financial backtest experiments, we analyze that the state transition probability is known in the portfolio, and value function can be directly obtained by sampling, further theoretically proving the optimality of the proposed reinforcement learning method in the portfolio. Finally, the superiority and generality of our approach are demonstrated through comprehensive experiments on the cryptocurrency dataset, S&P 500 stock dataset, and ETF dataset.

1. Introduction

Portfolio selection refers to dispersing the capital into various financial assets such as stocks, bonds, or cryptocurrencies in some periods, and continually adjusting the investment weight of each asset by analyzing its performance and correlation with other assets. The goal is to diversify investment risks and increase expected cumulative returns on investments. With the continuous development of the financial market, a wide range of financial assets has developed. Consequently, it has become increasingly difficult for individual investors to analyze each financial asset and the relationship among assets to make investment decisions. At the same time, investors may make irrational investment behavior (Hirshleifer & Shumway, 2003). It refers to the fact that affected by emotions, rumors and fraud, investors often make wrong judgments on financial assets (Kawy et al., 2021). Therefore they will formulate and implement irrational financial asset trading strategies (Liu et al., 2020). Herding behavior of the securities market is a typical example, which means investors blindly follow others in decision-making (Sharma & Bikhchandani, 2000). In recent years, due to the widespread application of artificial intelligence in the financial field, quantitative trading has developed rapidly in portfolio selection. Quantitative trading is defined as a procedure established by mathematical statistics based on certain rules, which can automatically trade assets in the financial market (Kawy et al., 2021). It can mine potential patterns of asset price changes, thereby constructing investment policy and automatically executing transactions. The characteristics of quantitative trading are highly automated and continuous (Liu et al., 2020). Compared with manual operation, it can efficiently collect information and place orders, while avoiding irrational trading behavior. In light of these advantages, the traditional inefficient manual trading pattern is shifting to an efficient quantitative trading pattern (Huang et al., 2019).

Since the asset price sequences in the financial market possess the characteristics of multi-noise, nonlinearity, and non-stationarity (Malkiel, 2003; Tsai & Hsiao, 2010). Due to the complexity and challenge of the problem, many existing quantitative methods such as Gao and Zhang (2013), Jiang et al. (2017), Li et al. (2013) have insufficient modeling ability for these complex asset price data. Therefore those methods cannot fully mine and utilize the effective information in them, resulting in the instability of their performance (Xu et al., 2021; Zhang et al., 2022). Consequently, in the current investment portfolio industry, more methods still need experienced investors' supervision

* Corresponding author at: School of Software, Shandong University, Jinan 250101, China.
E-mail addresses: tianlongzhao@mail.sdu.edu.cn (T. Zhao), xiangma@mail.sdu.edu.cn (X. Ma), xmli@sdu.edu.cn (X. Li), czhang@sdu.edu.cn (C. Zhang).

https://doi.org/10.1016/j.eswa.2023.119707
Received 29 March 2022; Received in revised form 4 February 2023; Accepted 12 February 2023
Available online 27 February 2023
0957-4174/© 2023 Elsevier Ltd. All rights reserved.

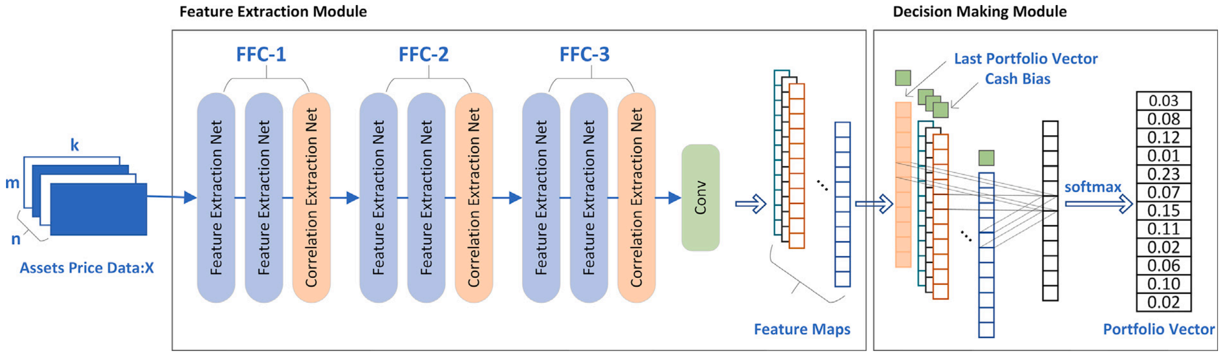


Fig. 1. The structure of the PolicyNet.

during its trading process. In this way, profitability losses caused by methods performance fluctuations can be prevented. Therefore, how to improve the shortcomings of the existing methods to establish a more accurate quantitative model has become an urgent problem to be solved in portfolio selection.

The quantitative approach to building a portfolio selection policy first requires mining sequential patterns for each asset. According to Technical Analysis (Dempster et al., 2001), which shows that asset price contains all information, the sequential pattern of each asset can be mined using the features extracted from asset price sequential data. The majority of the traditional methods for extracting the features are based on traditional technical indicators, such as moving average (Li et al., 2015), weighted moving average (Wang et al., 2018), MACD (Gunduz et al., 2017), etc. However, these standard manual indicators give insufficient consideration to the characteristics of different assets (Xu et al., 2021), so the extracted features have poor representation ability, resulting in poor profit results. For example, features extracted through moving averages can effective describe the trends characteristics of assets. However, when assets are in the mean-reversion market and have significant volatility characteristics, they may suffer significant losses using these features (Poterba & Summers, 1988). The rapid development of deep learning has led to several methods, such as RNN (Zaremba et al., 2014; Zhu et al., 2021), LSTM (Hochreiter & Schmidhuber, 1997), TCN (Bai et al., 2018), and their variants, showing strong representation ability with time series data and achieving great success in modeling time series data (Sutskever et al., 2014).

To build a portfolio selection policy also needs to consider the correlations among assets. The financial assets in the portfolio contain complex correlations that may change rapidly over time (Stefanova & Elkamhi, 2011). Previous researches (Frye, 2008; Lopez, 2004) have shown that assets correlations can help estimate investment risk in the portfolio and thus guide more effective portfolio management. For example, investors can hedge their investment risk in a bear market by diversifying their portfolios based on assets correlations (Tasca et al., 2017). In a bull market, they may increase portfolio returns by taking advantage of the synergy of upward/related assets (Xu et al., 2021). Traditional methods mainly focus on mining linear correlation (Györfi et al., 2008; Li et al., 2011) in assets, while existing deep learning-based methods (Xu et al., 2021; Zhang et al., 2022) are too simple for correlation modeling. These methods still have some shortcomings in mining and utilizing the complex nonlinear correlation (Chen et al., 2020) between assets. Since Google proposed the transformer model, the self-attention mechanism shows the strong performance on establishing the nonlinear correlations among data by using the multi-head structure and nonlinear activation function (Vaswani et al., 2017). But it uses point-wise connection for data and lacks modeling of temporal relationship of data. That leads to quadratic complexity of sequence length and lower temporal information utilization rate.

Based on the performance of the assets and the correlation among assets, the quantitative trading agents finally need to learn a reliable investment policy. Since the portfolio aims at maximizing cumulative return, the investment decision of each period should not only consider the return of the current period but also consider the impact of this decision on the future. Therefore, portfolio selection can be defined as a Markov Decision Process(MDP)which can be solved through Reinforcement Learning(RL). Deep Reinforcement Learning(DRL) method combined with Deep Learning has been studied extensively in the portfolio field (Deng et al., 2016; Wu et al., 2020). RL needs to estimate the value function using network estimation (Wu et al., 2020) or Monte Carlo sampling (Jiang et al., 2017) to guide policy update. However, the current studies (Jiang et al., 2017; Liang et al., 2018; Zhang et al., 2022) reflect that the reinforcement learning method based on the value function network performed poorly in the portfolio field but did not offer a penetrating explanation. Meanwhile, methods (Jiang et al., 2017; Zhang et al., 2022) based on Monte Carlo sampling use an approximate way to replace the theoretically recurrent structure in the experiment. Despite accelerating the speed of policy network training, it would achieve suboptimal performance.

For the shortcomings of the above works, this paper constructs a new policy network (TARN) to mine the asset price information in the portfolio and then output the investment decision, as shown in Fig. 1. First, TARN extracts the price feature of each asset using dilated causal convolution in TCN. Compared with RNN-based methods, TCN has the advantage of parallel computation and no problem with gradient disappearance (Bai et al., 2018). The recent work (Xu et al., 2021) uses point-wise product in self-attention to model the sequence information of a single asset, which has the problem of quadratic complexity of sequence length and lower temporal information utilization rate. Different from it, TARN performs new sequence-level dot product operation among assets to model assets correlations based on extracted features. The advantage of this way is not only to retain the temporal information of each asset, but also to reduce the time complexity of attention mechanism from $O(L^2)$ to linear $O(L)$. Then, we deeply analyze the knownness of the state transition probability in the portfolio backtest experiment, which infers that the true value function can be obtained by one-time sampling based on Monte Carlo. In this way, we give the theoretical proof of an important problem in the research field of portfolio (Jiang et al., 2017; Liang et al., 2018; Zhang et al., 2022): Why reinforcement learning methods based on Monte Carlo sampling to obtain value functions outperform those based on network to estimate value functions. Based on this proof, we construct the deterministic policy gradient recurrent reinforcement learning method based on once Monte Carlo sampling to train the policy network. In the backtest experiment, it is theoretically optimal compared with methods based on network to estimate value functions and methods canceling the recurrent structure. Experiments show that our method further improves the performance of the policy without increasing the generation time of the transaction decision.

Our main contributions are summarized as follows :

• A policy network based on the self-attention mechanism is proposed to model the nonlinear correlations among asset prices and to output portfolio selection decisions.

• A deterministic policy gradient recurrent reinforcement learning method based on Monte Carlo sampling is constructed to train the proposed policy network with the aim of maximizing cumulative return.

• The theoretically proof of an important problem in the research field of portfolio is given: Why reinforcement learning methods based on Monte Carlo sampling to obtain value functions outperform those based on network to estimate value functions.

• In the portfolio backtest experiment, the reinforcement learning method constructed in this paper is theoretically optimal compared with methods based on network to estimate value functions and methods canceling the recurrent structure.

• Through comprehensive contrast experiments on different datasets, the advancement and generalization of the method proposed in this paper are verified.

The article is organized as follows. In chapter 2, we review the existing portfolio selection methods. In chapter 3, we formalize the portfolio selection problem and use the formula to represent the whole process of portfolio selection. In chapter 4, the portfolio policy network based on asset correlation is introduced. In chapter 5, the deterministic policy gradient reinforcement learning method based on Monte Carlo sampling is introduced, and its optimality is proved. In chapter 6, we conduct experiments on cryptocurrency and stock datasets to verify the advantages and generalization of our method. In chapter 7, we summarize and prospect our work.

2. Related work

The research of portfolio selection can be divided into two schools, namely mean–variance theory (Markowitz, 1952) and capital growth theory (Kelly, 2011). The former focuses on single-period investment, whose goal is to balance the portfolio's return (mean) and risk (variance); the latter focuses on multi-period investment to maximize cumulative returns. Since portfolio focuses on long-term investments to overcome short-term fluctuations, the current mainstream research methods are based on capital growth theory. These mainstream methods can be divided into traditional methods and the latest reinforcement learning-based methods.

The traditional methods can be divided into four categories (Li & Hoi, 2014): following winners, following losers, pattern matching, and meta-learning. Following winners-(Agarwal et al., 2006; Gaivoronski & Stella, 2000) refers to investing capital in the past better performance assets, which is based on the investment concept of 'momentum effect'. Following losers (Li & Hoi, 2012; Li et al., 2012) refers to investing capital in the past pool performance assets, which is based on the investment concept of 'reversal effect'. Pattern matching (Györfi et al., 2006) predicts future asset price distribution based on historical asset price data distribution, which is based on the investment philosophy of 'history will reappear'. Different from the single investment strategy used in the above three categories, meta-learning (Das & Banerjee, 2011) integrates various investment strategies to generate new ones. Traditional methods mine asset price characteristics through manually constructed technical indicators and output investment decisions according to manual formulated trading rules. For different markets, different periods, and different assets, these methods trade based on fixed technical indicators and fixed rules. Therefore, the effectiveness of these methods depends on the technical indicators and rules whether they are suitable for the current market. However, fixed technical indicators and rules, which are not updated in the changing financial market, cannot always be effective in the dynamic portfolio environment (Deng et al., 2016).

Compared with traditional methods, deep reinforcement learning (Carta et al., 2021; Deng et al., 2016; Jiang et al., 2017; Park et al., 2020; Soleymani & Paquet, 2020; Wu et al., 2020; Xu et al., 2021; Zhang et al., 2022) can learn persistent and efficient investment policy from different markets, different periods, and different assets through end-to-end methods. Deep reinforcement learning contains three types, which are actor-based network (Wang et al., 2019), critic-based network (Yuan et al., 2019), and actor–critic-based network (Lowe et al., 2017). Those methods have been widely studied in portfolio selection, but their effects are uneven. Recent works (Chen et al., 2020; Jiang et al., 2017; Liang et al., 2018) shows that the critic-based network has poor performance and makes some guesses. In this paper, we analyze the reasons for the failure of the critic-based network in detail and demonstrate the optimality of the proposed reinforcement learning method.

In terms of modeling the correlations among assets based on the extracted features, the traditional methods mainly focus on linear correlation measurement, such as Euclidean distance (Györfi et al., 2008), correlation coefficient (Li et al., 2011), etc. Due to the extensive nonlinear correlation (Chen et al., 2020) among financial assets, these methods cannot accurately excavate those correlations. Most reinforcement learning methods only consider the price features of each asset, ignoring the price correlation between assets. The work Jiang et al. (2017) pioneers the Ensemble of Identical Independent Evaluators (EIIE) structure, using the identical network to extract the price sequence features of each asset separately. Most follow-up works (Park et al., 2020; Shi et al., 2019; Soleymani & Paquet, 2020) start on this basis, but none of these methods consider the correlation among assets. However, considering the price correlation among assets can better disperse risks and improve returns. The work Zhang et al. (2022) uses a simple convolution neural network, and the work Xu et al. (2021) uses self-correlation to model the correlation among assets. Both methods have achieved better results, indicating that asset price can positively affect portfolio selection. Nevertheless, these simple methods cannot sufficiently model the nonlinear correlation of assets. To solve this shortage, we adopt the self-attention mechanism to accurately model the nonlinear correlation through multi-head structure and nonlinear activation functions.

3. Formal expression of portfolio problem

In this section, the portfolio selection process is formalized during some periods. Cryptocurrency market is selected as the investment object, which can continuously conduct transactions within 24 hours. Like previous studies (Jiang et al., 2017; Park et al., 2020; Shi et al., 2019; Soleymani & Paquet, 2020), our investment goal is to earn more Bitcoin, which is the most recognized cryptocurrency. In other words, Bitcoin in our portfolio is essentially equivalent to cash in the stock market. To facilitate automated transactions, the investment time is divided into equal periods of 30 minutes, so the transaction frequency is 48 times a day.

3.1. Portfolio selection process

Assume $m+1$ assets are invested during n periods. The first asset is set to Bitcoin as the quoting currency, which means the price of other assets will be settled in Bitcoin. For asset i, four kinds of price are extracted from tth period as the representation of this period's price sequence: the highest price $v^h_{i,t}$, the lowest price $v^l_{i,t}$, the opening price $v^o_{i,t}$, and the closing price $v^c_{i,t}$. Therefore, all assets' price data of tth period is $\mathbf{x}_t = \left\{\mathbf{v}^o_t, \mathbf{v}^c_t, \mathbf{v}^h_t, \mathbf{v}^l_t\right\} \in \mathbb{R}^{(m+1)\times 4}$. And the price data of the last k periods are utilized to make investment decisions for tth period, namely $\mathbf{X}_t = \left\{\mathbf{x}_{t-k}, \dots, \mathbf{x}_{t-1}\right\} \in \mathbb{R}^{k\times(m+1)\times 4}$. Because Bitcoin is quoting currency, its four prices are constant to 1, namely $v^o_{0,t}=v^c_{0,t}=v^h_{0,t}=v^l_{0,t} = 1$. To represent the price change in tth period, the relative price $\mathbf{p}_t$ is defined as:

$$\mathbf{p}_t := \mathbf{v}_t^c \oslash \mathbf{v}_{t-1}^c = \left(1, \frac{v_{1,t}^c}{v_{1,t-1}^c}, \frac{v_{2,t}^c}{v_{2,t-1}^c}, \ldots, \frac{v_{m,t}^c}{v_{m,t-1}^c}\right)^\top. \tag{1}$$

Assume the portfolio weight at the beginning of tth period is $\mathbf{w}_t \in \mathbb{R}^{(m+1)\times 1}$, where $\sum_i w_{i,t} = 1$. Then, at the end of tth period, the portfolio weight change to

$$\mathbf{w}_t' = \frac{\mathbf{p}_t \odot \mathbf{w}_t}{\mathbf{p}_t{}^\top \mathbf{w}_t}, \tag{2}$$

where $\bigodot$ is the element-wise multiplication, $\mathbf{p}_t \odot \mathbf{w}_t$ represents the change of investment weight of each asset. At the beginning of (t+1)th period, the portfolio weight is adjusted to $\mathbf{w}_{t+1}$ according to the trading policy.

When $\mathbf{w}_t'$ is adjusted to $\mathbf{w}_{t+1}$ by buying and selling assets in the portfolio, the proportion of transaction cost is a scalar $c_{t+1} = f\left(\mathbf{w}_t', \mathbf{w}_{t+1}\right)$ calculated in Lowe et al. (2017) and the proportion of remaining total capital is $\mu_{t+1} = 1 - c_{t+1}$. Then the logarithmic return of (t+1)th period is:

$$r_{t+1} = \ln\left(\mu_{t+1}\mathbf{p}_{t+1}{}^\top \mathbf{w}_{t+1}\right). \tag{3}$$

In an investment process with transaction cost, n periods, and r_0 initial capital, the cumulative logarithmic return of the portfolio is:

$$R = r_0 \sum_{t=1}^{n} \ln\left(\mu_t \mathbf{p}_t^\top \mathbf{w}_t\right) = r_0 \sum_{t=1}^{n} r_t. \tag{4}$$

3.2. General assumption

In this paper, all experiments are conducted on backtest. Since the data in the backtest is known, referring to the most advanced and influential works (Jiang et al., 2017; Liu et al., 2020; Xu et al., 2021; Zhang et al., 2022), the following two hypotheses (Li & Hoi, 2012; Vajda, 2006) need to be satisfied:

1. **Full liquidity**: All transactions can be carried out immediately according to a given decision.

2. **Not affecting the market**: The result of the transaction will not impact the prices of the assets in the future market.

4. Portfolio policy network based on asset correlation

In order to improve the shortcomings of the existing methods in dealing with assets nonlinearity correlation, this paper proposes a policy network composed of dilated causal convolution network and self-attention mechanism. The network structure of this work is different from Zhang et al. (2022) which models asset correlations by one simple convolution network. Based on each asset's price features extracted by dilated causal convolution operation, the policy network uses the self-attention mechanism to model the nonlinear correlation among assets. Through end-to-end training, the policy network constantly updates parameters to follow the changing market. Thus, the continuously effective price features that include the correlation among assets are extracted. Finally, the policy network will output the portfolio decision according to these price features.

4.1. Structure of policy network

The policy network structure consists of the feature extraction module and the decision module, as shown in Fig. 1. In the feature extraction module, the policy network constructs three identical sub-modules (FFC^{1-3}) and serializes them to enhance the feature extraction ability. Each sub-module (FFC^l) comprises two price feature extraction networks and one price correlation extraction network in series. The price feature extraction network ($FENet$) is used to extract the price sequence features of each asset in the portfolio. The price correlation extraction network ($CENet$) is used to model the price correlation among assets based on each asset's price sequence features extracted by $FENet$. Also, some Relu activation functions, Dropout regularization, and normal convolution operations are set in the module, as shown in Fig. 2.

Residual connections are also set to solve the problem of multi-layer network training, allowing the network to focus only on the current differences. The output of the FFC^3 module is passed through a convolutional layer to obtain the final feature maps of the feature extraction module. Then, the decision module is used to process these feature maps and output the portfolio selection decision for each period.

4.2. Feature extraction of asset price sequence

According to the theory of technical analysis, the historical price of assets has a vital impact on the future. Therefore, how to extract the effective features of asset price sequences in the portfolio to guide portfolio selection is a significant issue. In this section,we build the same $FENet$ module as work (Zhang et al., 2022) and use the dilated causal convolution network (DCC) (Bai et al., 2018) to extract the price sequence features of each asset. Early works, such as the EIIE structure proposed in Jiang et al. (2017), use traditional convolutional networks to extract asset price sequence features. Because their convolution kernel size limitation cannot obtain long-term dependence information, they are not suitable for extracted long-time sequence information. Recently, researchers have developed recurrent neural networks, such as RNN and LSTM, which have better ability in modeling long-time, non-stationary and noisy sequential data. However, methods of this structure suffer from gradient vanishing issues and have limited parallel computing capabilities (Wang et al., 2021).

In comparison, the DCC solves the drawbacks of traditional convolutional networks and recurrent neural networks in time sequence information processing through causal convolution and dilated convolution. The causal convolution (Fig. 3(a)) avoids the input of neurons in the latter layer network receiving future information by filling and filter shifting. In addition to having similar function to recurrent neural networks, this structure also facilitates parallel computation and alleviates the vanishing problem. The dilated causal convolution(Fig. 3(b)) uses interval sampling to get a larger receptive field than the traditional convolution kernel, which can obtain long-term dependence information. In this way, the extracted long-term price sequence features of each asset are more effective. The operation of $FENet$ in FFC^l can be formulated as:

$$\mathbf{F}_{l,1}^p = Drop\left(Relu\left(DCC\left(\mathbf{F}_{l-1}\right)\right)\right) \tag{5}$$

$$\mathbf{F}_{l,2}^p = Drop\left(Relu\left(DCC\left(\mathbf{F}_{l,1}^p\right)\right)\right), \tag{6}$$

where $\mathbf{F}_0 = \mathbf{x}$ and DCC are the dilated causal convolution operation. Table 2 shows the specific dimension changes of feature tensor.

4.3. Modeling the correlation of price among assets

Portfolio selection not only needs to learn the historical price sequence characteristics of each asset but also needs to consider the price sequence correlation among assets to better hedge investment risks and improve returns.

There are many methods to extract assets correlations, such as fully connected layer, convolutional network, and self-correlation network. The difference between attention mechanism and the fully connected layer is that the attention mechanism can use input characteristics information to determine which parts of characteristics are more important (Vaswani et al., 2017). The representation of each asset characteristic takes into account other assets characteristics in the attention mechanism. And in a changing market, attention mechanism guarantees a higher weight when one asset is more important than other assets. However, in the full connection layer, the weight of each asset is

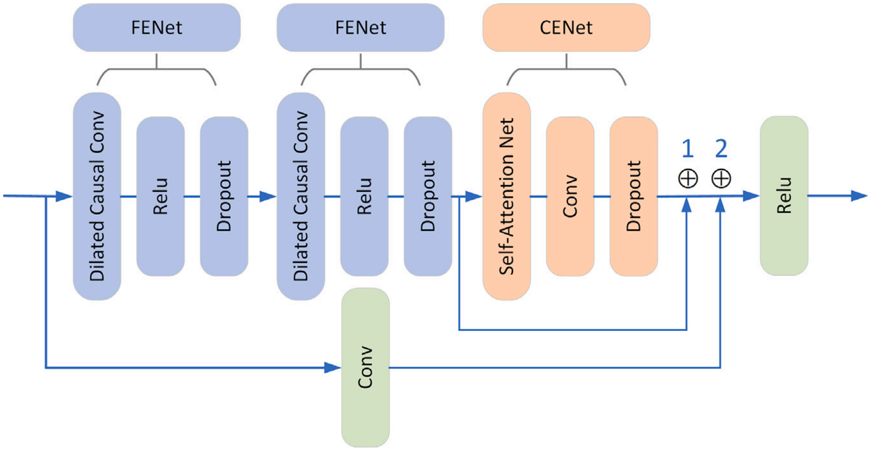
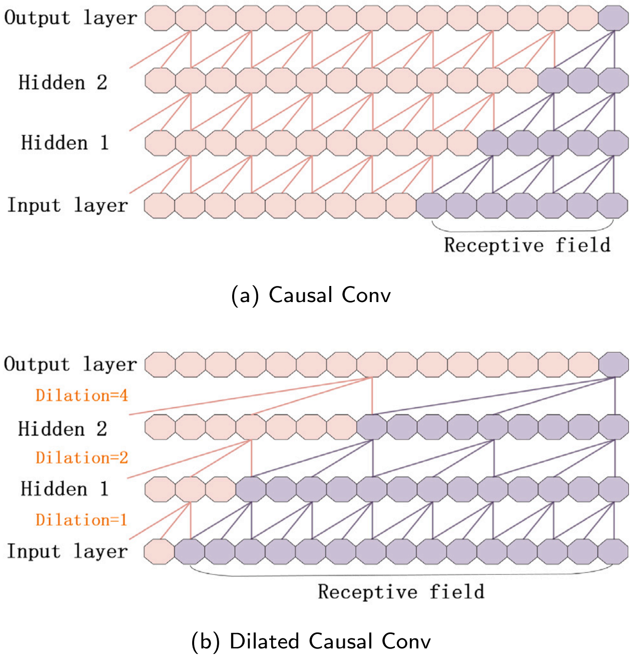


Fig. 2. The detail of the FFC.



(a) Causal Conv

(b) Dilated Causal Conv

Fig. 3. Comparison of Causal Conv and Dilated Causal Conv.

only related to the weight of its corresponding position in the network. When an asset performs better, it will reduce the return of the portfolio due to its previous low weight. Using convolutional network (Zhang et al., 2022) to extract assets correlations has a problem of the small receptive field. That is, the size of the convolution kernel is much smaller than the size of the input data. In this way, the assets in the portfolio cannot establish a comprehensive relationship with other assets, which affects the mining of correlations among assets. Compared with the self-attention mechanism, the self-correlation network (Xu et al., 2021) lacks the process of obtaining Q, K and V through three different linear transformations of the input. But Q, K and V introduce learnable parameters with a reasonable structure, so that the network has a stronger learning ability to capture more comprehensive context information. It also does not take the multi-head structure (Vaswani et al., 2017), which enables the model to learn different behaviors based on the same attention mechanism. Therefore, both methods have obvious deficiencies compared with the self-attention mechanism.

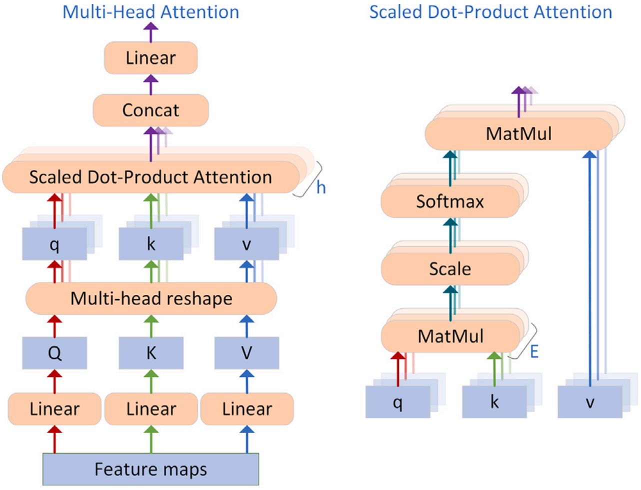


Fig. 4. Algorithm flow of the attention mechanism.

Unlike applying the attention mechanism on asset sequential data itself, applying the attention mechanism among assets has two advantages:1. The application of the point-wise product in the self-attention mechanism to assets sequential data will destroy the temporal information of the sequential data and reduce the temporal information utilization rate. Applying the attention mechanism among assets only performs sequence-level dot product operation among assets, preserving the data temporal information of each asset. 2. The complexity of the attention mechanism apply to asset sequential data itself is $O(m \times L^2)$. When apply among assets, the complexity turns to $O(m^2 \times L)$. Since the number of assets m is constant, both time complexities are asymptotically $O(L^2)$ and $O(L)$. Therefore, applying the attention mechanism among assets has a lower time complexity.

Therefore, we build the $CENet$ module and use the self-attention mechanism to encode the price sequence features extracted by the $FENet$. By utilizing the multi-head structure and nonlinear activation function in this mechanism, we can effectively establish the nonlinear correlations among assets. Through the softmax function and multi-head structure in this mechanism, it can model a more comprehensive nonlinear correlation of asset price and guide portfolio selection. The specific process is shown in Fig. 4, and the algorithm flow is below.

The algorithm flow of $CENet$ in FFC^l:

1. Firstly, different linear transformations are performed to obtain the query matrix **Q**, the key matrix **K**, and the value matrix **V** on the feature maps, which are the output of the DCC. Through matrix segmentation, $\mathbf{Q}$, $\mathbf{K}$ and $\mathbf{V}$ are mapped to h heads to obtain the multi-pair price correlation feature matrices $\mathbf{q}$, $\mathbf{k}$, and $\mathbf{v}$ among assets, as shown in the left half of Fig. 4. We formulate those linear transformations and multi-head operations as:

$$\mathbf{Q} = Liner^{q}\left(\mathbf{F}_{l,2}^{p}\right) \tag{7}$$

$$\mathbf{K} = Liner^{k}\left(\mathbf{F}_{l,2}^{p}\right) \tag{8}$$

$$\mathbf{V} = Liner^{v}\left(\mathbf{F}_{l,2}^{p}\right) \tag{9}$$

$$\left[\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_h\right] = MultiHead\left(\mathbf{Q}\right) \tag{10}$$

$$\left[\left(\mathbf{k}_1, \mathbf{v}_1\right), \left(\mathbf{k}_2, \mathbf{v}_2\right), \dots, \left(\mathbf{k}_h, \mathbf{v}_h\right)\right] = MultiHead\left(\mathbf{K}, \mathbf{V}\right) \tag{11}$$

2. Then the matrix $\mathbf{q}$ and $\mathbf{k}$ are multiplied in each head to get the attention distribution $\mathbf{E}$ of each asset relative to all assets. In order to prevent excessive inner product, each component $\mathbf{e}$ of $\mathbf{E}$ is divided by $\sqrt{d}$ for scaling, where d is the dimension of each component. After that, attention distribution $\mathbf{E}$ is standardized through the softmax function so that the sum of attention distribution of each asset relative to all assets is 1. Finally, the processed attention distribution in each head is multiplied by corresponding $\mathbf{v}$, as shown in the right half of Fig. 4. The scaled dot-product operation is formulated as:

$$\mathbf{E}_i = \frac{\mathbf{q}_i{}^{\top}\mathbf{k}_i}{\sqrt{d}} \tag{12}$$

$$\mathbf{M}_i = Softmax\left(\mathbf{E}_i\right)\mathbf{v}_i \tag{13}$$

3. The multiply result is contacted and then passed through a linear convolution layer. Finally, the attention value of each asset in each head relative to all assets is obtained as:

$$\mathbf{Attention} = Convolution\left(Contact\left(\mathbf{M}_1, \dots, \mathbf{M}_i, \dots, \mathbf{M}_h\right)\right). \tag{14}$$

Through the attention network, the output feature maps contain not only the price sequence features of each asset but also the price correlation among assets. Compared with the methods without considering the price correlation among assets or simply considering the price correlation among assets (Xu et al., 2021; Zhang et al., 2022), our method can capture more comprehensive features in the portfolio. The output of FFC^{l} is formulated as:

$$\mathbf{F}_l^{c} = drop\left(\mathbf{Attention}\right), \tag{15}$$

$$\mathbf{F}_l = Relu\left(\mathbf{F}_l^{c} \oplus \mathbf{F}_{l,2}^{p} \oplus Convolution\left(\mathbf{F}_{l-1}\right)\right), \tag{16}$$

where $\oplus$ means residual connection.

After modeling assets correlations through self-attention mechanism, the portfolio has the ability to hedge. When the price of some assets in the portfolio is in a relative downward trend, the weight of these assets in the attention distribution E will be reduced, and the investment funds for them will decrease. On the contrary, other assets with relatively rising or stable prices will increase investment weight and get more investment funds. Since the total amount of capital is fixed, the portfolio does not take more risk in the market but increases the expected return.

4.4. Decision model

As shown in Fig. 1, the decision-making module will make the final portfolio decision based on the asset price feature maps $\mathbf{F}$ extracted in series by three FFC sub-modules and a convolution layer. It can be formulated as:

$$\mathbf{F} = Convolution\left(\mathbf{F}_3\right). \tag{17}$$

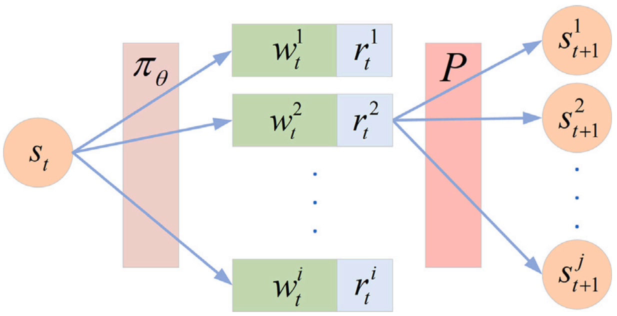


Fig. 5. The Markov decision process.

Considering that the excessive change of investment weight will bring high transaction costs, the previous investment weight is added in the feature maps to reduce the change of investment weight in the current period. This adding operation is formulated as:

$$\mathbf{F}' = \mathbf{F} + \mathbf{w}^{p}, \tag{18}$$

where + means to add one feature map(previous investment weight) into existing feature maps.

Meanwhile, considering that any asset investment will reduce the return of Bitcoin when the price of all assets falls relative to Bitcoin. In this case, we say the portfolio has been suffering an overall downward trend relative to Bitcoin. Therefore, adding Bitcoin bias to the feature maps as a risk-free asset is necessarily. This adding operation is formulated as:

$$\mathbf{F}'' = AD\left(\mathbf{F}', 1\right), \tag{19}$$

where AD means add one dimension with the value of 1 (Bitcoin is quoting currency)to each feature map in $\mathbf{F}'$.

After that, the fully-connected operation is performed on the feature maps, and its output is normalized by the softmax function (the sum of investment weights of each asset is guaranteed to be 1). Finally, the investment weights of different assets are output as:

$$\mathbf{w} = Softmax\left(FC\left(\mathbf{F}''\right)\right), \tag{20}$$

where FC is fully-connected operation.

5. Reinforcement learning framework

Existing financial asset price prediction and trend prediction work need to annotate unstable and noisy financial data, and use supervised learning to learn investment policy. Different from this, considering that portfolios pursue the maximization of cumulative returns, we convert the entire portfolio selection process into a **Markov Decision Process**. And we use a reinforcement learning algorithm to learn the policy network constructed in Chapter 4.

5.1. Markov decision process

The Markov decision process of portfolio selection is shown in Fig. 5. First, according to the current works (Jiang et al., 2017; Xu et al., 2021; Zhang et al., 2022), tth period's market state $\mathbf{s}_t$ is represented with the previous k periods' asset price $\mathbf{x}_t$ and (t-1)th period output action(investment weight) $\mathbf{w}_{t-1}$, namely $\mathbf{s}_t = \left(\mathbf{X}_t, \mathbf{w}_{t-1}\right)$.

In Fig. 5, the probability distribution of actions taken by the trading agent in a particular state is represented by:

$$\pi_\theta\left(\mathbf{w}_t \mid \mathbf{s}_t\right) = p\left[\mathbf{W} = \mathbf{w}_t \mid \mathbf{S} = \mathbf{s}_t\right], \tag{21}$$

where θ represents the parameters used to construct the policy. The logarithmic return of the trading agent after taking action $\mathbf{w}_t$ under state $\mathbf{s}_t$ is represented by:

$$r_t = \ln\left(\mu_t \mathbf{p}_t^{\top}\mathbf{w}_t\right). \tag{22}$$

After taking the action $\mathbf{w}_t$ of the tth period, the probability distribution $\mathbf{p}^{\mathbf{w}_t}_{\mathbf{s}_t,\mathbf{s}_{t+1}}$ of the environment moving to a new state $\mathbf{s}_{t+1}$ in the (t+1)th period is represented by:

$$\mathbf{P}^{\mathbf{w}_t}_{\mathbf{s}_t,\mathbf{s}_{t+1}} = p\left[\mathbf{S}' = \mathbf{s}_{t+1} \mid \mathbf{S} = \mathbf{s}_t, \mathbf{W} = \mathbf{w}_t\right]. \tag{23}$$

The expected return (cumulative return) obtained by using policy π_θ under state $\mathbf{s}_t$ is called the **state value function**, which is expressed as :

$$V_{\pi_\theta}\left(\mathbf{s}_t\right) = E_{\pi_\theta}\left[\sum_{i=t}^{n} r_i \mid \mathbf{S} = \mathbf{s}_t\right]. \tag{24}$$

The expected return (cumulative return) obtained after taking action $\mathbf{w}_t$ according to policy π_θ under state $\mathbf{s}_t$ as the **action value function**, which is expressed as:

$$Q_{\pi_\theta}\left(\mathbf{s}_t, \mathbf{w}_t\right) = E_{\pi_\theta}\left[\sum_{i=t}^{n} r_i \mid \mathbf{S} = \mathbf{s}_t, \mathbf{W} = \mathbf{w}_t\right]. \tag{25}$$

Reinforcement learning aims to learn the optimal policy to maximize the state value function and the action value function.

5.2. Optimality proof of deterministic policy gradient method based on Monte Carlo sampling

In order to estimate the action value function Q and learn the optimal policy π, reinforcement learning used in the portfolio usually adopts two types of methods: value function-based methods and model-free policy gradient-based methods. The methods based on value functions, such as Monte Carlo and Temporal-Difference, usually cannot deal with the problem of continuous state and action. Therefore, their application is limited in the financial field, which contains continuous prices and investment weights.

The method based on model-free policy gradient is divided into deterministic policy gradient (DPG) (Silver et al., 2014) and stochastic policy gradient (SPG) (Sutton et al., 2000) according to whether the action output by the Actor is determined. However, investment is more biased towards certainty in financial investment, so the DPG is more suitable. The method based on DPG usually adopts the Actor–Critic structure. The Actor is the policy that outputs the action according to the input state. The Critic estimates the action value function Q according to the input state and the Actor's output action to guide the Actor updating to the better policy. The estimation method for Critic can be divided into two kinds. One uses the Monte Carlo sampling method, through a large number of sampling using an empirical average to approximate the expected cumulative return; the other, such as DDPG (Lillicrap et al., 2015) and TD3 (Fujimoto et al., 2018), use the neural network method to input the current state and action to approximate the expected cumulative return. Many works (Liang et al., 2018; Zhang et al., 2022) had applied Critic-network to the portfolio field, but they had not achieved good returns. Next, we will analyze its reasons and prove that the true action value function can be obtained by Monte Carlo sampling compared to neural networks.

According to the conversion relationship between state value function and action value function (Eq. (10) and Eq. (11)), the Bellman equation of action value function can be deduced to Eq. (12).

$$V_{\pi_\theta}\left(\mathbf{s}_{t+1}\right) = \sum_{\mathbf{w}_{t+1}} \pi_\theta\left(\mathbf{w}_{t+1} \mid \mathbf{s}_{t+1}\right) Q_{\pi_\theta}\left(\mathbf{s}_{t+1}, \mathbf{w}_{t+1}\right) \tag{26}$$

$$Q_{\pi_\theta}\left(\mathbf{s}_t, \mathbf{w}_t\right) = r^{\mathbf{w}_t}_{\mathbf{s}_t} + \sum_{\mathbf{s}_{t+1}} P^{\mathbf{v}_t}_{\mathbf{s}_t \to \mathbf{s}_{t+1}} V_{\pi_\theta}\left(\mathbf{s}_{t+1}\right) \tag{27}$$

$$\begin{aligned} Q_{\pi_\theta}\left(\mathbf{s}_t, \mathbf{w}_t\right) ={}& r^{\mathbf{w}_t}_{\mathbf{s}_t} + \sum_{\mathbf{s}_{t+1}} P^{\mathbf{w}_t}_{\mathbf{s}_t \to \mathbf{s}_{t+1}} \sum_{\mathbf{w}_{t+1}} \pi_\theta\left(\mathbf{w}_{t+1} \mid \mathbf{s}_{t+1}\right) \cdot \\ & Q_{\pi_\theta}\left(\mathbf{s}_{t+1}, \mathbf{w}_{t+1}\right) \end{aligned} \tag{28}$$

In general, due to the unknowability of the environment, the state transition probability P is unknown. Even if the reward $r^{\mathbf{w}_t}_{\mathbf{s}_t}$ and action distribution $\pi_\theta\left(\mathbf{w}_{t+1} \mid \mathbf{s}_{t+1}\right)$ can be determined according to the policy π_θ, the action value function Q cannot be directly solved.

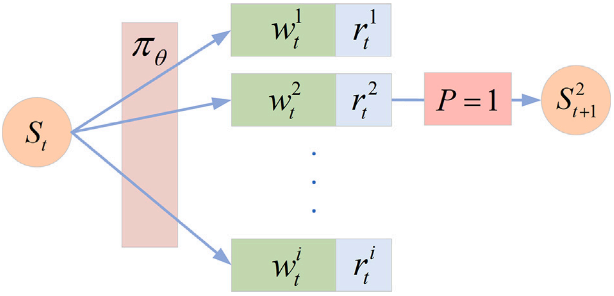


Fig. 6. The Markov decision process with the known state transition probability.

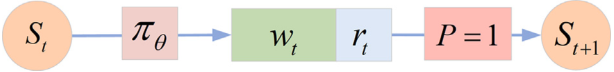


Fig. 7. The Markov decision process with the deterministic policy and the known state transition probability.

Therefore, in order to obtain Q and improve the policy, the Critic module can only estimate Q' to approximate the true Q, that is $Q'_{\pi_\theta}\left(\mathbf{s}_t, \mathbf{w}_t\right) \approx Q_{\pi_\theta}\left(\mathbf{s}_t, \mathbf{w}_t\right)$. In this way, the suboptimal policy π'_θ improved by estimating Q' can only approach the optimal policy π_θ, and the investment return $R\left(\pi'_\theta\right)$ obtained by the suboptimal policy is not higher than $R\left(\pi'_\theta\right)$ guided by the optimal policy, namely $R\left(\pi'_\theta\right) \leq R\left(\pi_\theta\right)$.

However, the state $\mathbf{s}_t$ of tth period in the portfolio is usually composed of the asset price $\mathbf{X}_t$ of previous k periods and the portfolio weight $\mathbf{w}_{t-1}$ of (t-1)th period. Under assumption 2, the results of all transactions will not affect the prices of assets in the market. Thus, as shown in Fig. 6, when the investment weight $\mathbf{w}_t$ for the tth period is determined by policy π_θ, the next state is known as $\mathbf{s}_{t+1} = \left(\mathbf{X}_{t+1}, \mathbf{w}_t\right)$, so the state transition probability of the environment is determined, and the probability $P^{\mathbf{w}_t}_{\mathbf{s}_t \to \mathbf{s}_{t+1}}$ of transition to state $\mathbf{s}_{t+1}$ in the (t-1)th period is 1. In this case, Eq. (12) becomes Eq. (13):

$$\begin{aligned} Q_{\pi_\theta}\left(\mathbf{s}_t, \mathbf{w}_t\right) ={}& r^{\mathbf{w}_t}_{\mathbf{s}_t} + \sum_{\mathbf{w}_{t+1}} \pi_\theta\left(\mathbf{w}_{t+1} \mid \mathbf{s}_{t+1}\right) \cdot \\ & Q_{\pi_\theta}\left(\mathbf{s}_{t+1}, \mathbf{w}_{t+1}\right) \end{aligned} \tag{29}$$

At the same time, according to the deterministic policy gradient theorem, the output action is unique in a particular state, namely $\pi_\theta\left(\mathbf{w}_{t+1} \mid \mathbf{s}_{t+1}\right) = 1$.

In this case, as shown in Fig. 7, Eq. (13) becomes Eq. (14), which is expressed as:

$$Q_{\pi_\theta}\left(\mathbf{s}_t, \mathbf{w}_t\right) = r^{\mathbf{w}_t}_{\mathbf{s}_t} + Q_{\pi_\theta}\left(\mathbf{s}_{t+1}, \mathbf{w}_{t+1}\right). \tag{30}$$

Then, the true action value function Q can be obtained through once Monte Carlo sampling under the determined state transition probability, which no longer need to take a large number of samples to estimate expectation and is represented by:

$$\begin{aligned} Q_{\pi_\theta}\left(\mathbf{s}_t, \mathbf{w}_t\right) &= E_{\pi}\left[\sum_{i=t}^{n} r_i \mid \mathbf{S} = \mathbf{s}_t, \mathbf{W} = \mathbf{w}_t\right] \\ &= \sum_{i=t}^{n} r_i \mid \mathbf{S} = \mathbf{s}_t, \mathbf{W} = \mathbf{w}_t. \end{aligned} \tag{31}$$

Therefore, the investment policy π^{MC}_θ learned based on the true action value function, which is obtained by only once Monte Carlo sampling, will be better than the policy $\pi^{\mathrm{Net}}_\theta$ learned by network estimation. The investment return under policy $\pi^{\mathrm{Net}}_\theta$ will not be higher than that guided by policy π^{MC}_θ, expressed as:

$$R\left(\pi^{\mathrm{MC}}_\theta\right) \geq R\left(\pi^{\mathrm{Net}}_\theta\right) \tag{32}$$

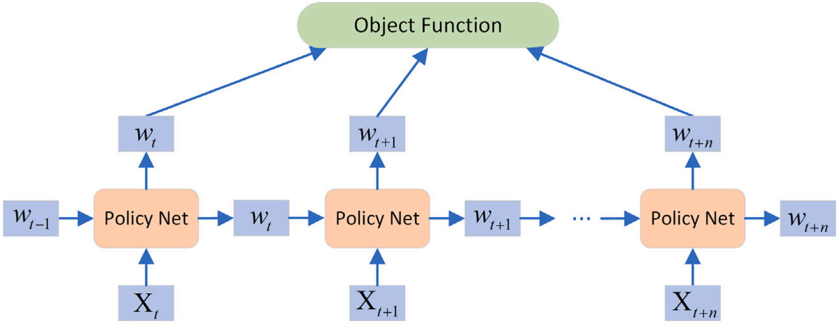


Fig. 8. The Deterministic Policy Gradient Recurrent Reinforcement Learning based on Monte Carlo Sampling.

According to Eq. (16), the DPG based on Monte Carlo sampling is superior to other methods based on unknown state transition probability in portfolio selection.

5.3. Deterministic policy gradient recurrent reinforcement learning based on Monte Carlo sampling

According to the above analysis, we use Monte Carlo sampling to obtain the true value function and then construct a recurrent deterministic policy gradient method (Fig. 8) to learn and update the policy network constructed in Chapter 4. the value function of Monte Carlo sampling, namely cumulative return R, is taken as the objective function:

$$R = r_0 \sum_{t=1}^{n} \ln\left(\mu_t \mathbf{p}_t{}^{\top} \mathbf{w}_t\right) = r_0 \sum_{t=1}^{n} r_t, \tag{33}$$

where r_0 represents the initial capital, r_t represents the return per period. In order to maximize the cumulative return of the portfolio, the policy network's parameter θ is constantly updated along the gradient direction of the objective function as below:

$$\theta \rightarrow \theta + \lambda \nabla_{\theta} R, \tag{34}$$

where λ is the learning rate.

Although the objective function designed only considers cumulative returns, volatility risk term has been implicitly limited by transaction costs, and retracement risk term has been implicitly modeled in the design of the policy network (in Section 4.4, we have set Bitcion bias).

Unlike other works (Jiang et al., 2017; Xu et al., 2021; Zhang et al., 2022) saving each period's network output from previous training batch as the each period's network input in next training batch. In each training batch, this recurrent structure takes the input of the latter period network as the output of the previous period network. It is worth noting that whether the recurrent structure in the reinforcement learning method is adopted will only affect the training of the policy network. When using the recurrent structure method in the training of the policy network, the input of the latter network needs to wait for the output of the former network, which increases the training time compared with the non-recurrent structure. However, when generating a transaction decision in a certain period, it is only necessary to input the needed data in a well-trained single network (Fig. 1), which is not related to whether the recurrent structure is adopted. Therefore, the reinforcement learning method in this paper only increases the training time of the network, but does not increase the time of decision generation when trading.

6. Experiment

In this section, the adopted datasets, evaluation metrics, comparison methods, and experimental setup are first introduced. Then the proposed method is evaluated from the following three aspects through ablation experiments and comparison experiments: 1. Profitability on real datasets; 2. Risk-adjusted profitability; 3. Performance under retracement risk.

Table 1
The cryptocurrency dataset ('Num':the number of the divided data).

Dataset	Assets	Training data		Testing data	
		Data range	Num	Data range	Num
Crypto-A	11	2015-07 to 2017-05	32220	2017-05 to 2017-07	2777
Crypto-B	11	2016-01 to 2017-11	32218	2017-11 to 2018-01	2776
Crypto-C	15	2016-07 to 2018-05	32174	2018-05 to 2018-07	2772

6.1. Dataset and preprocessing

With the rapid development of cryptocurrency in recent years, cryptocurrency has gradually become a new and popular investment product. Poloniex was born in 2014 and has become the world's leading cryptocurrency trading platform. Poloniex allows easy trading of mainstream cryptocurrencies while also providing historical prices of various cryptocurrencies. We use Poloniex's API[1] interface to extract different cryptocurrencies data from different periods. The cryptocurrency database can be got in .db format and connected through sqlite3. Datetime and panel in pandas are used to establish time index and save data. There are many kinds of cryptocurrencies. According to the method in the work Jiang et al. (2017), different number and time of cryptocurrencies with the largest trading volume in the previous month are selected to form an portfolio, as shown in Table 1. It is crucial to note that all cryptocurrencies prices are relative-price to Bitcoin since our goal is to acquire more Bitcoin.

Moreover, cryptocurrencies appear at different times and may lack some historical prices for new cryptocurrencies. To this end, the method in the work Jiang et al. (2017) is used to fill the vacancy price data. Since the return on investment depends on the relative price change of assets rather than the absolute price, the input price of each period is standardized by dividing the price of the 30th minute to obtain the relative price.

6.2. Evaluation metrics

Similar to the works Jiang et al. (2017), Xu et al. (2021), Zhang et al. (2022), three standard metrics are used to evaluate the performance of each method. The first is the cumulative portfolio return(APV) as follows:

$$\text{APV} = S_n = S_0 \prod_{t=1}^{n} \left(1 - c_t\right) \mathbf{p}_t{}^{\top} \mathbf{w}_t, \tag{35}$$

where initial capital S_0 set to 1, w_t is the portfolio weight vector determined at the beginning of period t, p_t is the relative price vector

[1] https://poloniex.com/support/api/

Table 2
Detailed network architecture of the portfolio policy network, using abbreviation: Conv: convolution layer; N: the number of output channel; K: kernel size; S: stride size; P: padding size; H: head number; Sc: scaled size; DiR: dilation rate; DrR: dropout rate; ⊕:residual connection.

Feature extraction module		
Architecture	Input Output shape	Layer Information
DDS-1	(11,31,4)→(11,31,8) (11,31,8)→(11,31,8) (11,31,8)→(11,31,8) (11,31,8)→(11,31,8)	DCC(N8, K[1x3], S1, P2), DiR1, DrR0.2,Relu DCC(N8, K[1x3], S1, P2), DiR1, DrR0.2, Relu SAC{Q,K,V}(N8,K1,S1,H2),Sc$\sqrt{4}$,Softmax,Conv(N8,K1,S1),DrR0.01,⊕1 Conv(N8,K1,S1),⊕2,Relu
DDS-2	(11,31,8)→(11,31,16) (11,31,16)→(11,31,16) (11,31,16)→(11,31,16) (11,31,16)→(11,31,16)	DCC(N16, K[1x3], S1, P4), DiR2, DrR0.2,Relu DCC(N16, K[1x3], S1, P4), DiR2, DrR0.2,Relu SAC{Q,K,V}(N16,K1,S1,H2),Sc$\sqrt{8}$,Softmax,Conv(N16,K1,S1),DrR0.01,⊕1 Conv(N8,K1,S1),⊕2,Relu
DDS-3	(11,31,16)→(11,31,16) (11,31,16)→(11,31,16) (11,31,16)→(11,31,16) (11,31,16)→(11,31,16)	DCC(N16, K[1x3], S1, P8), DiR4, DrR0.2,Relu DCC(N16, K[1x3], S1, P8), DiR4, DrR0.2,Relu SAC{Q,K,V}(N16,K1,S1,H2),Sc$\sqrt{8}$,Softmax,Conv(N16,K1,S1),DrR0.01,⊕1 ⊕2,Relu
Conv	(11,31,16)→ (11,1,16)	CONV(N16, K[1x31], S1, P0), ReLU
Decision making module		
Concat	(11,16)+(11,1)+(11,17)→ (12,17) (12,17)→(12,1)	Feature concatenation CONV-(N1, K1, S1, P0), Softmax

of period t, and c_t is the proportion of transaction cost when the investment weight adjusted. APV can evaluate the profitability of the method in the case of calculating transaction costs.

Although APV can intuitively evaluate the profitability of investment strategies, it ignores the risk. In order to avoid the huge potential losses caused by excessive risks, the Sharpe ratio (SR) is used to comprehensively consider the benefits and risks of investment policy as follows:

$$\mathrm{SR} = \frac{\text{Average } \left(r_t^c\right)}{\text{Standard Deviation } \left(r_t^c\right)} \tag{36}$$

where r_t^c represents the logarithmic return of the tth period considering transaction costs. A higher Sharp ratio means higher returns and lower risks.

Notwithstanding the Sharp ratio takes risk into account when measuring investment returns, it equally treats the rise and retracement of investment. However, in the financial market, the retracement of investment needs more attention because the retracement will lead to capital loss and cause the transaction cannot be carried out. To this end, the Calmar Ratio (CR) is employed to trade off the return and retracement of investment policy as:

$$\mathrm{CR} = \frac{S_n}{\mathrm{MDD}}, \tag{37}$$

where MDD is the Max Drawdown, indicating the range from the rising peak to the bottom.

$$\mathrm{MDD} = \max_{t \cdot \tau > t} \frac{S_t - S_\tau}{S_t} \tag{38}$$

In summary, in the three metrics of APV, SR, and CR, the larger the value, the stronger the policy's profitability in the face of the same risk.

6.3. Contrast method

The comparison of portfolio selection methods can be divided into two types. The first are the traditional methods. Previous influential work Li and Hoi (2014) divided traditional methods into categories according to the direction of investment weight transfer. As with other recent works Jiang et al. (2017), Zhang et al. (2022), we select some classical methods from different categories as representatives for experimental comparison. Its purpose is to show the overall performance of traditional methods. These categories and the representation methods are as follows:

Benchmark methods

UBAH: It equally spreads the total fund into the pre-selected assets and holds them to the end.

Best: It chooses the asset with the most APV over the back-test interval.

CRP: It rebalances the portfolio to a fixed investment proportion every period.

Following winners methods

UP (Cover, 1991): It assigns the capital to a single class of base experts, lets the experts run, and finally pools their wealth.

EG (Helmbold et al., 1998): It optimizes the investment weight to the optimal weight of the last period and adds some regular terms.

Following Losers methods

Anticor (Borodin et al., 2003): It makes bet on the consistency of positive lagged cross-correlation and negative auto-correlation to exploit the mean reversion property.

OLMAR (Li & Hoi, 2012): It proposes a multiple period mean reversion, which explicitly predicts the next price vector as the moving average within a window.

Meta-Learning methods

ONS (Agarwal et al., 2006): It solves the optimization problem in EG with L2-norm regularization via online convex optimization technique.

These methods build financial models manually based on given rules, and they can also use some machine learning techniques to determine parameters. The performance of these methods depends on the effectiveness of the rules used in the market.

Others are model-free and unsupervised machine learning methods using advanced deep reinforcement learning(DRL) to deal with financial market transactions, such as EIIE (Jiang et al., 2017), PPN (Zhang et al., 2022), and RAT (Xu et al., 2021). These methods use the non-recurrent reinforcement learning method of deterministic policy gradient to learn the policy. They do not depend on the fixed rules and directly learn the investment policy from the input historical price data. The attention-based recurrent neural network (TARN) proposed in this paper is compared with the above traditional and advanced methods to verify the effectiveness of our method.

6.4. Experimental setup

Although Fig. 1 shows the overall structure of the policy network, to make the details of the network clearer, the structure settings are exhaustively described in Table 2. First, the asset number is set to 11(for Crypto-A and Crypto-B) and 15(for Crypto-C). The purpose is to verify whether our method has performance difference compared to other methods over different time spans and different asset number. The windows size k is set to 31, and the input price data is $\mathbf{X} \in \mathbb{R}^{(11/15)\times 33\times 4}$. Then the batch size is selected to 32, that is, to build 32 shared weight policy networks through the recurrent connection.

Table 3
The Performance comparisons in ablation experiment on the cryptocurrency datasets.

Method	Crypto-A			Crypto-B			Crypto-C		
	APV	SR(%)	CR	APV	SR(%)	CR	APV	SR(%)	CR
TN	54.57	8.30	197.36	4.80	3.91	6.85	129.27	15.76	898.90
TAN	71.28	8.58	279.64	5.50	4.15	8.83	147.59	16.15	1184.59
TRN	88.07	8.90	350.64	5.14	4.06	6.97	148.55	16.15	1292.83
TARN	**123.13**	**8.97**	**418.81**	**6.84**	**4.86**	**13.41**	**212.33**	**17.16**	**1664.07**

In addition, the proposed method is implemented on a single NVIDIA RTX 2080s GPU using the Tensorflow framework. At the same time, to verify that our method can achieve superior performance without requiring particular hyperparameter settings for each data set, we use the same hyperparameter in three data sets. The learning rate is set to 0.0028, the transaction cost in Poloniex is 0.25%, the number of training is 80000, the optimizer is Adam, the cash bias term is fixed as 0, and the initial investment weight is evenly distributed on 11 or 15 assets. In order to obtain stable results, the random seed is fixed to 0, and the final network training time is about 6.5 GPU hours.

6.5. Ablation experiment

In order to verify the effectiveness of asset correlation based on attention mechanism to extract price features and recurrent reinforcement learning method to learn policy network, three ablation experiments are set up:

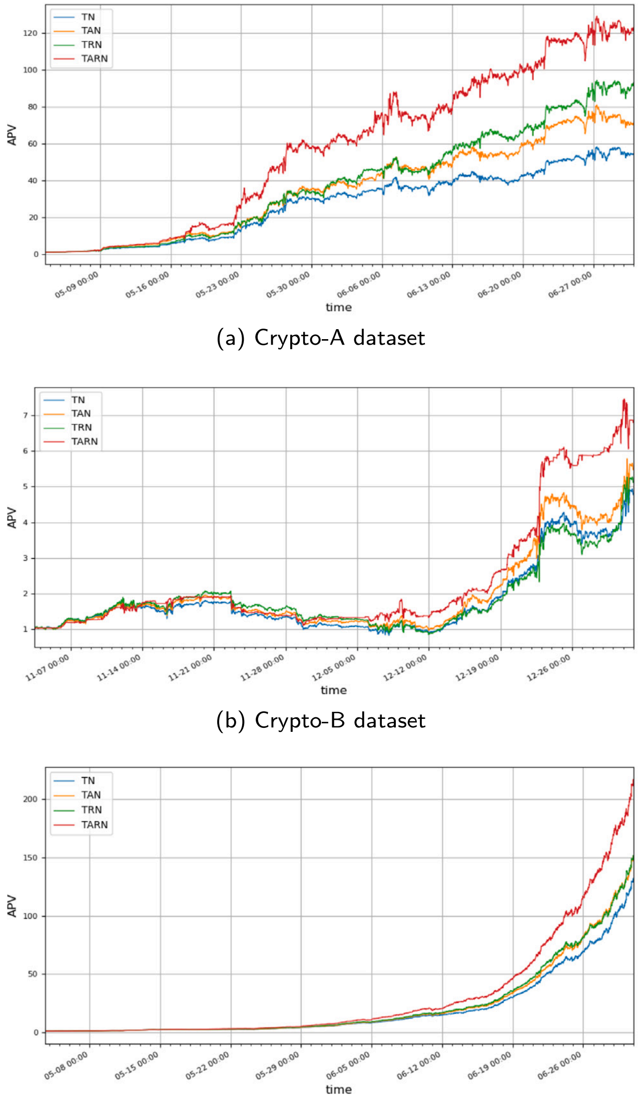
TN: Compared with TARN, it removes the self-attention network and adopts an investment policy that does not consider the asset correlation. At the same time, the network training method is replaced by the deterministic policy gradient non-recurrent reinforcement learning method, which is used in works (Jiang et al., 2017; Xu et al., 2021; Zhang et al., 2022). This contrast method is used to verify the joint effectiveness of attention mechanism and recurrent reinforcement learning;

TAN: Compared with TARN, it uses the attention network to consider the asset correlation but is trained based on non-recurrent reinforcement learning. This contrast method is used to verify the learning ability of the recurrent reinforcement learning method on the policy network;

TRN: Compared with TARN, it is trained based on recurrent reinforcement learning but does not use the attention network to consider the asset correlation. This method is used to verify the ability of the attention network to extract price features.

The return curves for TARN and its three degenerate variants in the test set are shown in Fig. 9. The metrics of APV, SR, and CR are used to reflect the modeling ability of the two parts in asset price data, as shown in Table 3.

It can be seen from Fig. 9 and Table 3 that the APV, SR, and CR values of TARN are higher than those of three degenerated variants, especially the values of APV and CR. Comparing the three metrics of TARN and TAN, TRN and TN, it can be seen that updating the parameters of the policy network through recurrent reinforcement learning, whether or not the attention network is used to extract the asset correlation, can significantly improve the performance of the investment policy; comparing the three indicators of TARN and TRN, TAN and TN, it can be seen that the use of attention network to extract asset correlation, whether or not training the policy network through recurrent reinforcement learning, can also improve the performance of the investment policy; finally, through the common comparison of TARN, TAN, TRN, and TN, it can be found that based on asset correlation modeling by attention network, training the policy network by recurrent reinforcement learning can bring more remarkable improvement to the performance of investment policy. It verifies the effectiveness of asset correlation modeling through attention mechanism and training policy network by recurrent reinforcement learning proposed in this paper.



(a) Crypto-A dataset

(b) Crypto-B dataset

(c) Crypto-C dataset

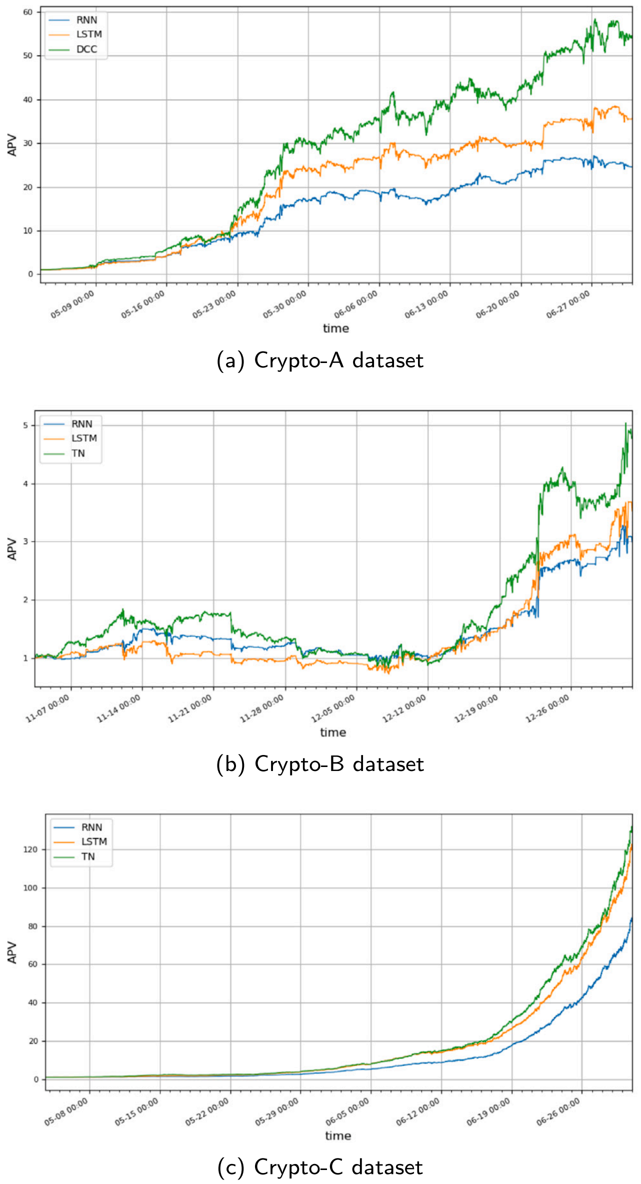
Fig. 9. The return curve comparisons in ablation experiment on the cryptocurrency datasets.

6.6. Asset price sequences feature extraction ability

The purpose of this section was to verify the relative effectiveness of using DCC to extract asset price sequences features. Therefore, we conducted a comparative experiment to compare the performance of different methods for extract asset price sequence features. In order to conduct this experiment, we replace the DCC network of the TN variant with RNN and LSTM. The performance will be reflected in the investment return evaluation metrics based on the extracted features. The return curves are shown in Fig. 10, and the APV, SR, and CR metrics values are shown in Table 4.

Table 4
The performance comparisons in ablation experiment on the cryptocurrency datasets.

Method	Crypto-A			Crypto-B			Crypto-C		
	APV	SR(%)	CR	APV	SR(%)	CR	APV	SR(%)	CR
RNN	24.62	7.72	104.39	2.99	3.16	4.60	82.58	15.23	757.25
LSTM	35.67	7.83	123.01	3.53	3.45	5.71	120.09	**16.16**	**1219.19**
TN	**54.57**	**8.30**	**197.36**	**4.80**	**3.91**	**6.85**	**129.28**	15.76	898.90



(a) Crypto-A dataset

(b) Crypto-B dataset

(c) Crypto-C dataset

Fig. 10. The return curve comparison in feature extraction experiment on the cryptocurrency datasets.

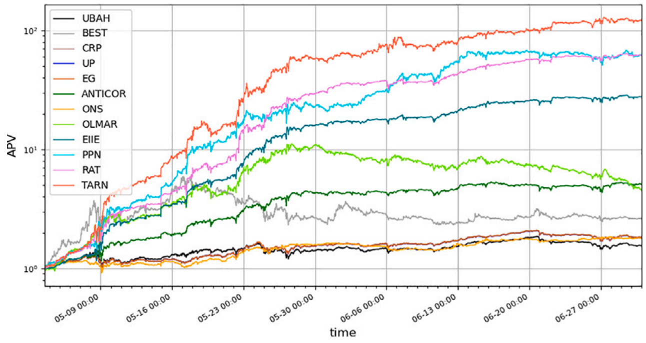


Fig. 11. The return curve comparison on the Crypto-A dataset.

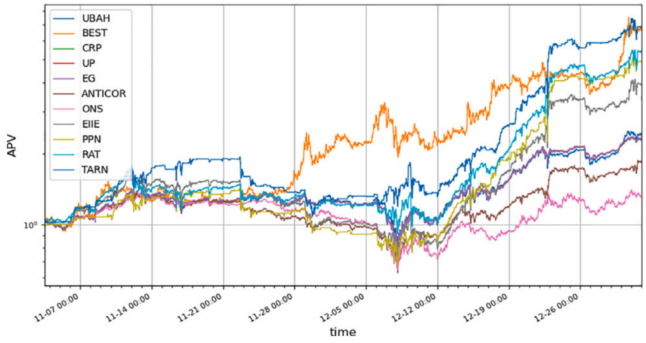


Fig. 12. The return curve comparison on the Crypto-B dataset.

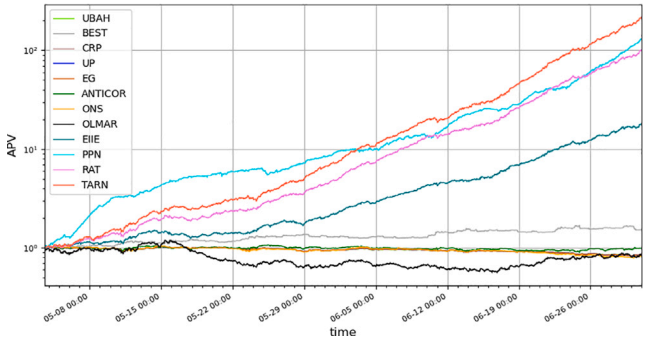


Fig. 13. The return curve comparison on the Crypto-C dataset.

According to Fig. 10. and Table 4, RNN as the feature extraction method of asset price series has achieved the worst performance in return evaluation metrics. This result can be explained by the gradient vanishing caused by the recurrent structure. As a consequence, the LSTM method, which alleviates gradient vanishing problem via multiple gating units, performs better than RNN in terms of return evaluation metrics. In contrast, the dilated causal convolution network does not adopt the recurrent structure to avoid the inherent gradient vanishing problem. By this way, DCC method achieves the best performance of APV in all test sets, as well as the highest returns across almost all investment periods. It also well done in other evaluation metrics in most cases, although they are not the primary consideration.

6.7. Profitability comparison

In order to verify the advancement of our method, TARN is compared with eight traditional methods and three advanced methods in three datasets. The return curves are shown in Figs. 11–13, and the APV, SR, and CR metrics values are shown in Table 5.

We first analyze the test sets of three datasets. Each evaluation metric of the Uniform Buy And Hold (UBAH) method represents the

Table 5
The performance of different methods on the cryptocurrency datasets.

Method	Crypto-A			Crypto-B			Crypto-C		
	APV	SR(%)	CR	APV	SR(%)	CR	APV	SR(%)	CR
UBAH	1.55	2.24	2.05	2.38	3.82	3.22	0.84	−2.3	−0.83
BEST	2.64	2.69	2.7	6.65	4.59	**13.67**	1.52	3.37	4.96
CRP	1.83	3.41	3.57	2.27	3.89	2.85	0.85	−2.29	−0.79
UP	1.81	3.33	3.47	2.29	3.90	2.89	0.85	−2.29	−0.79
EG	1.82	3.36	3.5	2.28	3.90	2.87	0.85	−2.29	−0.79
Anticor	5.10	5.55	21.9	1.84	2.54	1.56	0.98	0.06	−0.13
ONS	1.79	3.31	3.66	1.31	1.47	0.58	0.82	−2.27	−0.75
OLMAR	4.58	3.66	6.04	0.08	−3.30	−0.95	0.84	0.00	−0.29
EIIE	27.71	8.48	128.81	3.92	3.67	5.06	17.79	11.49	96.30
PPN	72.95	8.66	237.49	4.90	4.11	7.29	131.48	**19.07**	859.14
RAT	62.14	8.96	244.84	5.35	4.35	9.03	99.21	15.43	723.18
TARN	**123.13**	**8.97**	**418.81**	**6.84**	**4.86**	13.41	**212.33**	17.16	**1664.07**

overall market condition in a given investment period. Among them, APV value represents the market as a whole going up or down relative to Bitcoin(APV> or APV<1), SR and CR values approximate reflect the degree of market price volatility relative to Bitcoin(the smaller the value, the larger the volatility).

Since the APV of test sets A and B is greater than 1, the market in these two periods as a whole goes up. In the meantime, because the SR and CR values of test set A are smaller than those of test set B, the previous market is more volatile than the later. On the other hand, the APV value of test set C is less than 1, which shows that the market in this period is in an overall downward trend. Meanwhile, smaller SR and CR values in test set C indicate higher market volatility. With the above analysis, we summarize the market state of the three test sets: **market return**:B>A>C, **market volatility**:(A and C) > B.

By comparing the return curves and evaluation metrics in three test sets, we can see that the investment policy based on deep reinforcement learning is significantly superior to the traditional rule-based method. In particular, TARN has achieved excellent performance over different periods with different numbers of assets, proving the general effectiveness of our approach. We first analyze the reasons for the poor performance of traditional methods and put them down to three points.

First of all, no matter what types of traditional methods, they all have one thing in common. They are all based on predetermined rules, which, different from reinforcement learning methods, are not based on the current market. For example, the following winners methods always assign a higher investment weight to assets that have down well before, while the opposite is true for following losers methods. The former methods yield poor profits than the latter methods when the market is highly volatile: assets rising in price before are falling in price later. More precisely, in test set A with high volatile, the earning power of methods UP and EG is lower than that of methods Anticor and OLMAR. But the opposite is true for test set B, which has low volatility. Similarly, when the market continues to fall, like test set C, latter methods remain invested in assets that are falling in price, resulting in principal losses. Therefore, these methods can only perform better in the market that matches them. But in fact, the market is constantly changing, and these rules are to fail easily in the changed market, resulting in the loss of profits and even principal. Second, most traditional approaches do not take into account the correlation between assets. So they cannot use correlations to hedge their portfolios in a bear market, and cannot enhance portfolio returns in a bull market with the synergy of correlations. Third, different from reinforcement learning methods, most traditional methods only select the closing price as the input price feature of the model, and do not take into account the opening price, the highest price, the lowest price and other important price features. As the amount of modeling data becomes less, the model cannot accurately analyze the market, thus reducing the return on investment.

In contrast, when the market fluctuates greatly(test sets A and C), the method based on deep reinforcement learning achieves better returns than the traditional method, regardless of the overall upward or downward trend of the market. This study demonstrates that the investment policy generated by deep reinforcement learning can obtain greater benefits in the volatility. As discussed above, by effectively modeling the correlations among assets, TARN can take advantage of the relative changes of assets to hedge risks and carry out arbitrage when the market is volatile, so as to reduce risks and earn more profits than others. In relatively stable market(test set B), an investment strategy cannot make differential gains by buying assets with low valuations and selling assets with high valuations. Therefore, it is understandable why the method based on deep reinforcement learning do not perform as well as it do on the other two test sets, although TARN still achieves the best performance compared with traditional methods. However, when the market goes down, the investment return of traditional methods decreases, and even the principal is lost. This result further illustrates that the traditional rule-based method cannot be sustainable effective in the changing financial market(goes up or down), and cannot take advantage of volatility to get higher returns. On the contrary, the deep reinforcement learning method, which continuously updates the model based on the time series price data of the market, is able to continuously apply to market changes, makes effective decisions, and earns more profits.

At the same time, among the methods based on deep reinforcement learning, TARN has significantly improved in various metrics, especially APV and CR. The reasons are as follows: 1. EIIE does not consider the impact of the correlation among assets in the portfolio on investment returns; 2. Although PPN uses the convolution network to model the asset correlation, it cannot extract the comprehensive nonlinear correlation of assets. 3. RAT directly applies the Transformer method to the portfolio field, but the Transformer is proposed in NLP, which is not comprehensively compatible with the portfolio problem. It also uses linear autocorrelation to model the correlation of assets which cannot mine the nonlinear correlation among assets. 4. The above methods are based on degenerate deterministic policy gradient reinforcement learning, which does not adopt the non-recurrent network structure.

As mentioned in Section 5.3, although we do not define risk terms in the objective function, we implicitly take into account volatility and retracement risks through transaction costs and policy network design. Therefore, SR and CR values in TARN also achieved the best results in most cases, as shown in the results.

6.8. *Transaction cost rate sensitivity*

Transaction cost is one of the most important factors affecting investment returns. In Section 6.7, the TARN is verified that it is superior to other investment policies at a transaction cost rate of 0.25%. In order to verify that TARN has the same excellent profitability under different transaction cost rates, our method is compare with three advanced methods in different transaction cost rates. And the performance of profitability is listed in Table 6.

Table 6
The return performance of advanced methods under different transaction cost rates on the Crypto-A dataset.

Cost	0.01%	0.05%	0.10%	0.25%	1%
EIIE	42858.13	10342.8	1629.54	41.16	1.62
PPN	19809.65	8432.91	1910.36	72.95	2.14
RAT	69055.2	14362.64	1848.5	62.14	1.74
TARN	**75467.92**	**15641.19**	**2182.03**	**123.13**	**2.22**

Table 7
The S&P 500 stock dataset ('Num':the number of the divided data).

Dataset	Assets	Training data		Testing data	
		Data range	Num	Data range	Num
S&P500	506	2013-02 to 2017-08	1101	2017-08 to 2018-02	94

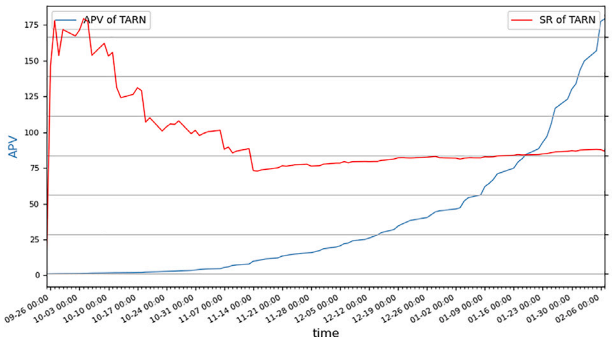


Fig. 14. The APV and SR curves of TARN on the S&P 500 dataset.

It can be seen from Table 6 that transaction cost rate have a crucial impact on the profitability of the policy. The higher the transaction cost rate is, the lower the profitability of the policy is. However, compared to other methods, TARN achieves the best APV performance at different transaction cost rates, which further confirms that TARN is more profitable than other methods.

6.9. *Performance in stock portfolio*

The above experiments verified that TARN has better performance on earning Bitcoin than other methods in the cryptocurrency dataset. However, to further verify the generalization performance of TARN, the same experimental settings are chosen on the S&P 500 stock dataset (Table 7) for profit comparative experiment. The S&P 500 dataset we use is a public dataset obtained from Kaggle.[2] It can be got in .csv format and connected through read_csv in pandas. Drop_duplicates and panel in pandas are used to establish stock/time indexes and save data. Fig. 14 shows the APV and SR curves on the stock dataset, and Table 8 shows the performance metrics comparison with other methods on stock dataset.

From Table 8, it can be found that traditional methods such as UBAH, CRP, etc. still achieve low performance and are weaker than methods based on deep reinforcement learning in various evaluation metrics. This result further confirms our statement in introduction: These standard manual indicators do not take into account different assets' characteristics, so the extracted features have poor representation ability, resulting in poor profit results. Also, the Best method refers to investing in an asset that will yield the best return in the future. But we found that deep learning-based approaches, especially ours, produced better returns than investing in the single highest-yielding stock. It further verifies the advantage of using deep reinforcement learning in the field of investment portfolio.

Table 8
The performance of different methods on the S&P 500 dataset.

Algos	UBAH	BEST	CRP	UP	EG
APV	1.21	89.63	1.21	1.22	1.22
CR	2.39	1417.49	2.34	2.32	2.34
SR(%)	12.03	10.37	11.69	11.7	11.73
Anticor	OLMAR	EIIE	PPN	RAT	**TARN**
1.09	17.81	99.35	167.84	112.47	**179.45**
0.89	208.32	1994.89	6792.51	151827.01	**INF**
5.85	78.04	108.31	148	**168.22**	155.22

Through Fig. 14 and Table 8, it is also show that the TARN has strong performance on the stock dataset. Compared with other methods in Table 8, TARN is still in the leading position on APV. More importantly, with no single day losing during 2017–08 to 2018–02, the drawdown in the whole process of investment guided by TARN is 0. Therefore CR metric cannot be calculated according to Eq. (37) and Eq. (38), which can only represent by INF. This result further proved that TARN is superior to other methods on the retraction risk item. It is also noteworthy that the SR value of RAT is higher than that of TARN, which indicates that TARN has a higher volatility relative to RAT. Taking into account the highest return and the zero drawdown of TARN, it can be concluded that relatively high volatility is what brings more returns to the portfolio when there is no retracement risk. From Fig. 14, we can find that SR curve is in a highly fluctuating state at the beginning of investment, and becomes stable with the extension of investment time. This shows that TARN can obtain a more stable risk-adjusted return rate over a longer investment time span. It shows the importance of long-term investment. Thus, experimental results on stock dataset show that TARN has strong generalization performance.

6.10. *Performance in ETF portfolio*

Similar to the stock dataset, we conduct comparative experiments on the advanced ETF dataset to further verify the performance of our model. From the financial market data website Tushare,[3] we obtain seven ETFs' historical market data and form a portfolio. They are selected from the dominant broad-based index ETFs in the Chinese stock market. And the codes of them are:510050, 159915, 513500, 510880, 159905, 159920, 159941. The ETF dataset spans from 2017-01 to 2022-07. The return curves are shown in Fig. 15, and the APV, SR, and CR metrics values are shown in Table 9.

From Fig. 15, we can find that on the ETF dataset, the return curves of all methods are relatively stable. The possible reason for this is that each asset in the portfolio is a broad-based index ETF (index fund). Thus, when they form a portfolio, they are equivalent to funds of funds (FOF). So the data in dataset is more stable than individual stocks. TARN still shows superior return performance in Fig. 15. PPN produces a significant retracement, and EIIE fails on the ETF dataset as well as the traditional approaches. Both reasons could be due to their inadequate asset correlation modeling capability.

As can be seen from Table 9, most of the methods achieve poor performance on the ETF dataset, leading to the loss of principal. The reason should be that the Chinese stock market has fallen significantly in the past two years, which makes the overall profit-making effect of the market weak. The only three profitable methods are all based on deep reinforcement learning, once again proving its superiority in the portfolio field. Among them, our method achieves the best results in APV and CR metrics, and is second only to RAT method in SR metric. On the whole, these results demonstrate the superiority of TARN over other methods on the ETF dataset. It reveals the effectiveness of asset correlation modeling through attention mechanism and training policy network by recurrent reinforcement learning proposed in this paper.

[2] https://www.kaggle.com/camnugent/sandp500

[3] https://www.tushare.pro/

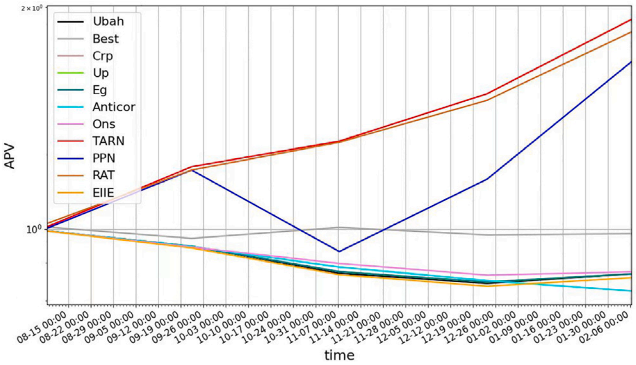


Fig. 15. The return curve comparison on the ETF dataset.

Table 9
The performance of different methods on the ETF dataset.

Algos	UBAH	BEST	CRP	UP	EG
APV	0.89	1.01	0.88	0.88	0.88
CR	−0.67	0.11	−0.68	−0.68	−0.69
SR(%)	−9.49	1.67	−9.59	−9.55	−9.59
Anticor	ONS	EIIE	PPN	RAT	**TARN**
0.83	0.89	0.87	1.98	2.01	**2.10**
−0.93	−0.76	−0.69	3.12	21.08	**25.34**
−14.27	−8.67	−9.67	21.24	**49.19**	46.23

7. Conclusion

This paper proposes an asset investment policy network derived from the self-attention mechanism. It can model the correlation among assets based on the features extracted from each asset price in the portfolio. At the same time, a deterministic policy gradient recurrent reinforcement learning method based on Monte Carlo sampling is constructed to train the policy network. It uses the objective function of maximum cumulative return. We prove the optimality of our reinforcement learning method in the portfolio field. Besides, our method has achieved the best performance on the most important metrics of portfolio cumulative return (APV), compared with traditional and advanced methods in comprehensive aspects of all datasets. Except in a few cases, the Sharpe ratio (SR) and Calmar Ratio (CR) of our method also have achieved the best performance in most cases. Overall, our method can result in a superior investment policy, based on the correlation among assets through self-attention mechanism modeling and deterministic policy gradient recurrent reinforcement learning. It can earn higher returns while suppressing risks.

In the future work, we will consider combining financial news (Li et al., 2021), to mine the correlation between text information and asset price sequence information. In this way, we can hopefully construct a better portfolio selection policy and guide investment.

CRediT authorship contribution statement

Tianlong Zhao: Software, Investigation, Writing – original draft. **Xiang Ma:** Software, Visualization. **Xuemei Li:** Visualization, Investigation. **Caiming Zhang:** Conceptualization, Methodology, Writing – review & editing.

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

The data that has been used is confidential.

Acknowledgments

This work are supported by National Natural Science Foundation of China (NSFC) (NO.62072281, NO.62007017).

References

Agarwal, A., Hazan, E., Kale, S., & Schapire, R. E. (2006). Algorithms for portfolio management based on the newton method. In *Proceedings of the 23rd international conference on machine learning* (pp. 9–16).

Bai, S., Kolter, J. Z., & Koltun, V. (2018). An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. arXiv preprint arXiv:1803.01271.

Borodin, A., El-Yaniv, R., & Gogan, V. (2003). Can we learn to beat the best stock. *Advances in Neural Information Processing Systems*, *16*.

Carta, S., Ferreira, A., Podda, A. S., Recupero, D. R., & Sanna, A. (2021). Multi-DQN: An ensemble of deep Q-learning agents for stock market forecasting. *Expert Systems with Applications*, *164*, Article 113820.

Chen, W., Jiang, M., & Jiang, C. (2020). Constructing a multilayer network for stock market. *Soft Computing*, *24*(9), 6345–6361.

Cover, T. M. (1991). Universal portfolios. *Mathematical Finance*, *1*(1), 1–29.

Das, P., & Banerjee, A. (2011). Meta optimization and its application to portfolio selection. In *Proceedings of the 17th ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 1163–1171).

Dempster, M. A., Payne, T. W., Romahi, Y., & Thompson, G. W. (2001). Computational learning techniques for intraday FX trading using popular technical indicators. *IEEE Transactions on Neural Networks*, *12*(4), 744–754.

Deng, Y., Bao, F., Kong, Y., Ren, Z., & Dai, Q. (2016). Deep direct reinforcement learning for financial signal representation and trading. *IEEE Transactions on Neural Networks and Learning Systems*, *28*(3), 653–664.

Frye, J. (2008). Correlation and asset correlation in the structural portfolio model. *The Journal of Credit Risk*, *4*(2), 75–96.

Fujimoto, S., Hoof, H., & Meger, D. (2018). Addressing function approximation error in actor-critic methods. In *International conference on machine learning* (pp. 1587–1596). PMLR.

Gaivoronski, A. A., & Stella, F. (2000). Stochastic nonstationary optimization for finding universal portfolios. *Annals of Operations Research*, *100*(1), 165–188.

Gao, L., & Zhang, W. (2013). Weighted moving average passive aggressive algorithm for online portfolio selection. In *2013 5th international conference on intelligent human-machine systems and cybernetics, Vol. 1* (pp. 327–330). IEEE.

Gunduz, H., Yaslan, Y., & Cataltepe, Z. (2017). Intraday prediction of Borsa Istanbul using convolutional neural networks and feature correlations. *Knowledge-Based Systems*, *137*, 138–148.

Györfi, L., Lugosi, G., & Udina, F. (2006). Nonparametric kernel-based sequential investment strategies. *Mathematical Finance: An International Journal of Mathematics, Statistics and Financial Economics*, *16*(2), 337–357.

Györfi, L., Udina, F., & Walk, H. (2008). Nonparametric nearest neighbor based empirical portfolio selection strategies. *International Mathematical Journal for Stochastic Methods & Models*, *26*(2), 145–157.

Helmbold, D. P., Schapire, R. E., Singer, Y., & Warmuth, M. K. (1998). On-line portfolio selection using multiplicative updates. *Mathematical Finance*, *8*(4), 325–347.

Hirshleifer, D., & Shumway, T. (2003). Good day sunshine: Stock returns and the weather. *The Journal of Finance*, *58*(3), 1009–1032.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, *9*(8), 1735–1780.

Huang, B., Huan, Y., Xu, L. D., Zheng, L., & Zou, Z. (2019). Automated trading systems statistical and machine learning methods and hardware implementation: A survey. *Enterprise Information Systems*, *13*(1), 132–144.

Jiang, Z., Xu, D., & Liang, J. (2017). A deep reinforcement learning framework for the financial portfolio management problem. arXiv preprint arXiv:1706.10059.

Kawy, R. A., Abdelmoez, W. M., & Shoukry, A. (2021). Financial portfolio construction for quantitative trading using deep learning technique. In *International Conference on Artificial Intelligence and Soft Computing* (pp. 3–14). Springer.

Kelly, J. L., Jr. (2011). A new interpretation of information rate. In *The kelly capital growth investment criterion: theory and practice* (pp. 25–34). World Scientific.

Li, B., & Hoi, S. C. H. (2012). On-line portfolio selection with moving average reversion. In *Proceedings of the 29th international conference on international conference on machine learning* (pp. 563–570).

Li, B., & Hoi, S. C. (2014). Online portfolio selection: A survey. *ACM Computing Surveys*, *46*(3), 1–36.

Li, B., Hoi, S. C., & Gopalkrishnan, V. (2011). Corn: Correlation-driven nonparametric learning approach for portfolio selection. *ACM Transactions on Intelligent Systems and Technology*, *2*(3), 1–29.

Li, B., Hoi, S. C., Sahoo, D., & Liu, Z.-Y. (2015). Moving average reversion strategy for on-line portfolio selection. *Artificial Intelligence*, *222*, 104–123.

Li, B., Hoi, S. C., Zhao, P., & Gopalkrishnan, V. (2013). Confidence weighted mean reversion strategy for online portfolio selection. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, *7*(1), 1–38.

Li, Q., Tan, J., Wang, J., & Chen, H. (2021). A multimodal event-driven LSTM model for stock prediction using online news. *IEEE Transactions on Knowledge and Data Engineering*, *33*(10), 3323–3337.

Li, B., Zhao, P., Hoi, S. C., & Gopalkrishnan, V. (2012). PAMR: Passive aggressive mean reversion strategy for portfolio selection. *Machine Learning*, *87*(2), 221–258.

Liang, Z., Chen, H., Zhu, J., Jiang, K., & Li, Y. (2018). Adversarial deep reinforcement learning in portfolio management. arXiv preprint arXiv:1808.09940.

Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., & Wierstra, D. (2015). Continuous control with deep reinforcement learning. arXiv preprint arXiv:1509.02971.

Liu, Y., Liu, Q., Zhao, H., Pan, Z., & Liu, C. (2020). Adaptive quantitative trading: An imitative deep reinforcement learning approach. In *Proceedings of the AAAI conference on artificial intelligence,Vol. 34, no. 02* (pp. 2128–2135).

Lopez, J. A. (2004). The empirical relationship between average asset correlation, firm probability of default, and asset size. *Journal of Financial Intermediation*, *13*(2), 265–283.

Lowe, R., Wu, Y., Tamar, A., Harb, J., Abbeel, P., & Mordatch, I. (2017). Multi-agent actor-critic for mixed cooperative-competitive environments. arXiv preprint arXiv:1706.02275.

Malkiel, B. G. (2003). The efficient market hypothesis and its critics. *Journal of Economic Perspectives*, *17*(1), 59–82.

Markowitz, H. M. (1952). Portfolio selection. *The Journal of Finance*, *7*(1), 77.

Park, H., Sim, M. K., & Choi, D. G. (2020). An intelligent financial portfolio trading strategy using deep Q-learning. *Expert Systems with Applications*, *158*, Article 113573.

Poterba, J. M., & Summers, L. H. (1988). Mean reversion in stock prices: Evidence and implications. *Journal of Financial Economics*, *22*(1), 27–59.

Sharma, S., & Bikhchandani, S. (2000). *Herd behavior in financial markets: A review, Vol. 00, no. 48*: *IMF working papers*, (p. 1).

Shi, S., Li, J., Li, G., & Pan, P. (2019). A multi-scale temporal feature aggregation convolutional neural network for portfolio management. In *Proceedings of the 28th ACM international conference on information and knowledge management* (pp. 1613–1622).

Silver, D., Lever, G., Heess, N., Degris, T., Wierstra, D., & Riedmiller, M. (2014). Deterministic policy gradient algorithms. In *International conference on machine learning* (pp. 387–395). PMLR.

Soleymani, F., & Paquet, E. (2020). Financial portfolio optimization with online deep reinforcement learning and restricted stacked autoencoder—DeepBreath. *Expert Systems with Applications*, *156*, Article 113456.

Stefanova, D., & Elkamhi, R. (2011). Dynamic correlation or tail dependence hedging for portfolio selection. In *AFA 2012 chicago meetings paper*.

Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to sequence learning with neural networks. In *Advances in neural information processing systems* (pp. 3104–3112).

Sutton, R. S., McAllester, D. A., Singh, S. P., & Mansour, Y. (2000). Policy gradient methods for reinforcement learning with function approximation. In *Advances in neural information processing systems* (pp. 1057–1063).

Tasca, P., Battiston, S., & Deghi, A. (2017). Portfolio diversification and systemic risk in interbank networks. *Journal of Economic Dynamics & Control*, *82*, 96–124.

Tsai, C.-F., & Hsiao, Y.-C. (2010). Combining multiple feature selection methods for stock prediction: Union, intersection, and multi-intersection approaches. *Decision Support Systems*, *50*(1), 258–269.

Vajda, I. (2006). Analysis of semi-log-optimal investment strategies. In *Prague stochastics* (pp. 719–727). Citeseer.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. In *Advances in neural information processing systems* (pp. 5998–6008).

Wang, Z., Huang, B., Tu, S., Zhang, K., & Xu, L. (2021). DeepTrader: A deep reinforcement learning approach for risk-return balanced portfolio management with market conditions embedding. In *Proceedings of the AAAI conference on artificial intelligence, Vol. 35, no. 1* (pp. 643–650).

Wang, H., Wu, Y., Min, G., Xu, J., & Tang, P. (2019). Data-driven dynamic resource scheduling for network slicing: A deep reinforcement learning approach. *Information Sciences*, *498*, 106–116.

Wang, Q., Xu, W., & Zheng, H. (2018). Combining the wisdom of crowds and technical analysis for financial market prediction using deep random subspace ensembles. *Neurocomputing*, *299*, 51–61.

Wu, X., Chen, H., Wang, J., Troiano, L., Loia, V., & Fujita, H. (2020). Adaptive stock trading strategies with deep reinforcement learning methods. *Information Sciences*, *538*, 142–158.

Xu, K., Zhang, Y., Ye, D., Zhao, P., & Tan, M. (2021). Relation-aware transformer for portfolio policy learning. In *Proceedings of the twenty-ninth international conference on international joint conferences on artificial intelligence* (pp. 4647–4653).

Yuan, Y., Yu, Z. L., Gu, Z., Yeboah, Y., Wei, W., Deng, X., Li, J., & Li, Y. (2019). A novel multi-step Q-learning method to improve data efficiency for deep reinforcement learning. *Knowledge-Based Systems*, *175*, 107–117.

Zaremba, W., Sutskever, I., & Vinyals, O. (2014). Recurrent neural network regularization. arXiv preprint arXiv:1409.2329.

Zhang, Y., Zhao, P., Li, B., Wu, Q., Huang, J., & Tan, M. (2022). Cost-sensitive portfolio selection via deep reinforcement learning. *IEEE Transactions on Knowledge and Data Engineering*, (1), 34.

Zhu, Q., Zhou, X., Tan, J., & Guo, L. (2021). Knowledge base reasoning with convolutional-based recurrent neural networks. *IEEE Transactions on Knowledge and Data Engineering*, *33*(5), 2015–2028.