Cross-insight Trader: A Trading Approach Integrating Policies with Diverse Investment Horizons for Portfolio Management

list of authors

Abstract—**Deep reinforcement learning (RL) has emerged as a promising approach for portfolio management due to its ability to make sequential decisions. However, applying RL techniques to this domain is still challenging due to the non-stationary nature of financial markets. To overcome this challenge, we propose cross-insight trader, a novel two-step RL-based approach that integrates multiple trading policies with different investment horizons to adapt to the changing market conditions. In the first step, we learn multiple horizon-specific policies by providing each policy with tailored information specific to its investment horizon. This allows each policy to recognize dynamic patterns within its respective horizon and make insightful pre-decisions. In the second step, we learn a cross-insight policy to make the final trade decision by considering the investment pre-decisions made by multiple horizon-specific policies in the first step. To enable effective learning of each policy, our approach employs a centralized critic to evaluate the actions performed by both horizon-specific and cross-insight policies. By incorporating multiple insights from different investment horizons into the decision-making process, our approach enhances its adaptability to changing market conditions. Experimental results conducted on three stock markets demonstrate the superiority of our framework.**

Index Terms—**Portfolio management, deep reinforcement learning, counterfactual mechanism, financial time series.**

I. Introduction

Portfolio management (PM), aiming to dynamically allocate wealth across a set of assets to maximize the investment return, is a fundamental yet challenging problem across several research communities, including finance, statistics and artificial intelligence [1]. Existing methods to deal with this task include online learning-based and reinforcement learning (RL)-based methods. Online learning-based methods (e.g., OLMAR [2] and Anticor [3]) use handcrafted features, such as moving average [4] and stochastic technical indicators [5] to optimize the allocation of wealth across a set of assets. Efforts have also been made to adopt deep reinforcement learning techniques to address the portfolio management problem [6]–[8]. Compared with online learning-based methods, RL is considered as a more promising technique for portfolio management since it can directly output the trading decisions through trial-and-error interactions with the financial environment to maximize the investment return. Despite efforts that have been made, applying RL to portfolio management still faces huge challenges because the non-stationary nature of the financial markets is difficult to be modeled by a single policy.

The non-stationary nature of financial markets can be attributed to different groups of investors operating with distinct investment horizons and trading strategies, as explained by the economic theory known as the fractal market hypothesis (FMH) [9]. FMH applies concepts from chaos theory to investment and economics, providing a framework to understand the dynamic nature of markets and the complex interactions among different market participants [10]. According to FMH, the financial market consists of various investors with different investment insights, who focus on different information, e.g., long-term investors focus on long-term horizon information and short-term investors focus on short-term horizon information.

Many works have studied how to alleviate this non-stationary problem in the RL scenario. Some pre-processing techniques, such as moving average [4] and wavelet transform [11] are adopted to smooth price series [12], [13]. Ensembling multiple policies is indeed a promising solution to enhance the robustness of models in portfolio management, and it is widely adopted in the financial industry [7], [14]. However, many existing RL-based methods for policies ensembling do not consider the existence of different investment horizon information in the financial market. This oversight is a limitation because it overlooks the diverse investment information prevalent in the financial environment. To make effective trading decisions, traders need to comprehensively consider the information provided by different investment horizons, as each horizon provides unique insights and patterns.

Motivated by the above intuitions, we present our novel cross-insight trader, a two-step approach that is designed to incorporate different investment insights simultaneously during the trading decision-making process. In the first step, we decompose the original price series into distinct sub-series, each containing specific information relevant to the corresponding investment horizon. These sub-series are then fed to the corresponding horizon-specific policies, enabling them to learn the dynamic patterns and make insightful pre-decisions accordingly. In the second step, we introduce a cross-insight policy that makes the final trade decision by considering the investment pre-decisions made by the horizon-

Z. Zheng, J. Shao, S. Deng, A. Zhu, F. Chen and H. T. Shen are with the School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, 611731, China. Email: {tzheng, dengshilon, anjiezhu} @std.uestc.edu.cn, {shaojie,chenfeiyu}@uestc.edu.cn, shenhengtao@hotmail.com. J. Shao, F. Chen and and H. T. Shen are also with Sichuan Artificial Intelligence Research Institute, Yibin, 644000, China.

Corresponding author: Jie Shao.

This work was supported by the National Natural Science Foundation of China (grants No. 62276047 and No. 61832001).

specific policies. We employ a centralized critic to evaluate the decisions of all policies. Specifically, the cross-insight policy is directly optimized based on the Q-value estimated by the critic. However, optimizing the horizon-specific policies solely based on the Q-value can be ineffective, because the Q-value only considers rewards obtained from the trade decision, and the gradient computed by the Q-value does not explicitly reason about how that particular horizon-specific policy's decision contributes to that reward. To address this limitation, we introduce a counterfactual mechanism that calculates a unique advantage for each horizon-specific policy by reasoning its contribution to the reward, thereby enhancing the training effectiveness of each horizon-specific policy.

In summary, our contributions are three-fold:

- To address the non-stationary nature of financial markets, we present a novel cross-insight portfolio management framework that takes into account different investment horizons when making trading decisions. This approach is unique and, to the best of our knowledge, has not been previously explored in the literature.
- We design a counterfactual mechanism based on theoretical derivation in our framework to achieve the contribution deduction to the final decision.
- Extensive experimental results on three real-world stock market datasets demonstrate that our framework achieves significant improvement compared with state-of-the-art approaches.

The remainder of the paper is organized as follows. We briefly introduce related work in Section II. Section III introduces the preliminaries about the task of portfolio management. After that, Section IV presents our proposed framework. The experimental evaluations are reported in Section V. Finally, this study is concluded in Section VI.

II. Related Work

We classify the existing portfolio management methods into two categories: online learning-based and reinforcement learning-based methods.

A. Online Learning-based Methods

The goal of the online learning-based methods is to maximize the expected log-return during sequential decision-making. Pioneering studies such as exponential gradient (EG) [15], [16], online Netwon step (ONS) [2], M0 [17], constant rebalanced portfolios (CRP) [18], anti-correlation (Anticor) [3], and universal portfolios (UP) [19] have been proposed to achieve a high-profit investment. Helmbold et al. [15] propose the EG strategy that incorporates the relative entropy as the regularization term. The optimization target of EG is focusing on the stock with the best performance in the last period but keeping the new portfolio close to the previous one to reduce the transaction cost. Agarwal et al. [2] propose ONS, which is similar to EG but replaces the relative entropy regularization term with L2-norm regularization and optimizes the model via online convex optimization technique. CRP [18] is a more straightforward strategy, which rebalances the portfolio to a fixed portfolio every period. UP [19] initially invests a proportion of wealth to each portfolio manager who runs the CRP strategy and lets the managers run it. In the end, each manager's wealth grows. Finally, the UP strategy pools the wealth of individual managers over these portfolio strategies. Different from the no distributional assumption in UP, the Anticor strategy [3] assumes that the market follows the mean reversion principle. To exploit the mean reversion, it statistically bets on the consistency of positive lagged cross-correlation and negative auto-correlation.

There are some other methods that also exploit the mean reversion to select the portfolio, such as passive aggressive mean regression (PAMR) [20], confidence weighted mean reversion (CWMR) [21], online moving average reversion (OLMAR) [4] and robust median reversion (RMR) [22]. PAMR [20] aims to design a loss function to reflect the mean reversion property, i.e., if the expected return based on the last relative price is greater than a threshold, the loss will increase linearly; otherwise, the loss will be zero. CWMR [21] further utilizes second-order portfolio information, i.e., the variance of the portfolio weights, follows the mean reversion principle, and learns online through confidence weighted. Unlike PAMR and CWMR that implicitly assume single-period mean reversion, OLMAR [4] exploits the multiple-period mean reversion to avoid one failure case on the real dataset. Observing that PAMR, CWMR and OLMAR suffer from estimation errors as they do not consider noises and outliers in the data, Huang et al. [22] propose RMR to exploit mean reversion via a robust L1-median estimator. However, all the above methods rely heavily on the handcrafted features, e.g., stochastic technical indicators or moving average. As a result, these models suffer from poor generalization ability and cannot achieve stable profits in the non-stationary markets.

B. Reinforcement Learning-based Methods

Reinforcement learning-based methods employ reinforcement learning to optimize specific trading policy and achieve the reward maximization [13], [23]–[25]. Some studies [26]–[29] incorporate deep learning to extract features in order to better represent the portfolio and optimize the investment return with RL. The popular one is the ensemble of identical independent evaluations (EIIE) [26]. EIIE implements the feature extraction by using three identical deep learning neural networks, which are a convolutional neural network (CNN), a basic recurrent neural network (RNN), and a long short-term memory network (LSTM). Xu et al. [27] propose relation-aware transformer to capture both sequential patterns and asset correlations for portfolios and exploit the reinforcement learning algorithm to train the model. Zhang et al. [28] propose a two-stream portfolio policy network to extract both asset correlations and sequential patterns, and devise a new cost-sensitive reward function to maximize the accumulative return via reinforcement learning. Deng et al. [29] implement a complex neural network to learn the asset features automatically and then the RL module could make trading decisions according to these features to achieve the maximization of reward. Some studies such as DeepTrader [8] and SARL [6] also take into account additional factors,

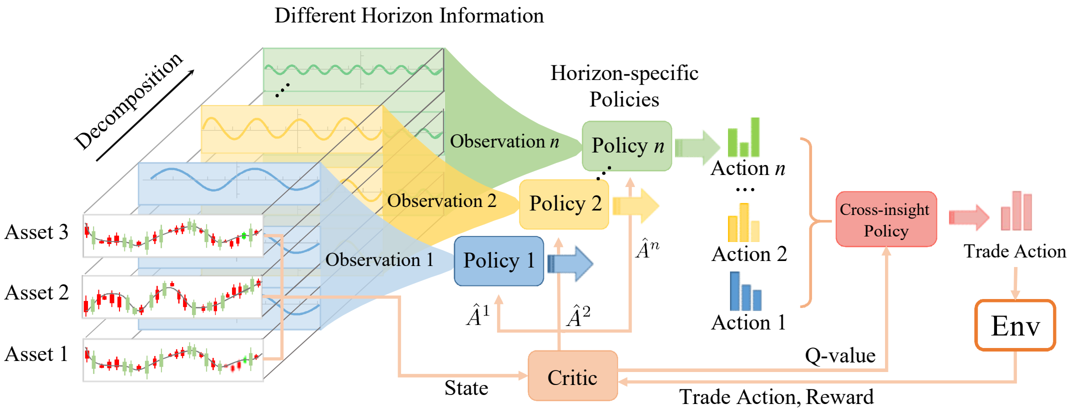


Fig. 1. Illustration of our proposed framework. $\hat{A}^k$ indicates the advantage for policy k reasoned by the counterfactual mechanism. Env represents the financial environment.

including financial news and market conditions, when making investment decisions. However, these models overlook the distinct insight information contained within the original stock data. Lee et al. [14] introduce MAPS, a cooperative multi-agent portfolio management system, where each agent aims to maximize its own reward by acting differently from others guided by different loss functions. Similarly, Niu et al. [7] propose MetaTrader, which incorporates an imitation objective into the RL framework to diversify the performance of various policies. However, unlike our framework, MAPS and MetaTrader primarily focus on distinguishing the behavior of each agent by utilizing a loss function when exposed to the same price series. Our work emphasizes incorporating diverse investment horizons and leveraging their distinct understandings of the market. By considering a broader range of insights, our framework provides a more comprehensive and insightful trading decision for portfolio management.

III. Problem Formulation

Portfolio management aims at reallocating the proportion of wealth over m assets in the financial market in order to maximize the investment return for a sequence of T trading time steps. Each asset has d kinds of features, such as opening, highest, lowest and closing prices when d=4. Let $p_{t,i} \in \mathrm{R}^d$ denote the price of asset i at time step t, where $t \in \{1, 2, ..., T\}$ and $i \in \{1, 2, ..., m\}$. We also define $P_t = \{p_{t,1}, p_{t,2}, ..., p_{t,m}\} \in \mathrm{R}^{m\times d}$ and $\mathcal{P}_t = \{P_{t-z}, ..., P_{t-1}\} \in \mathrm{R}^{z\times m\times d}$ to be the prices for all assets at time step t and the price series of previous z-moment prices regarding time step t. In our work, we only consider the closing prices, and thus d=1.

We formulate the PM task as a generalized Markov decision process (MDP) as many other works do [27], [28], [30]. An MDP can be defined as a tuple (S, A, Z, R), where $S = \{s_1, ..., s_T\}$ is a finite set of current states, $A = \{a_1, ..., a_T\}$ is a set of actions, Z is a stochastic state transition function determining the transition from current state s_t to next state s_{t+1} given the action $a_t \in A$, denoted as $s_{t+1} \sim Z(s_t, a_t)$, and $R = \{r_1, ..., r_T\}$ is a series of rewards received from the environment given the action. Note that in this work, we assume that the next state is determined by the market and the trading of our framework will not affect the state transition of each stock, i.e., $s_{t+1} \sim Z(s_t)$. In the context of PM, an MDP can be defined as follows:

- State s: a set of features that describe the current state of m assets. In general, various types of information, such as trading volumes, historical prices, sentiment scores, and financial statements, can be utilized as the current state of the assets. In our study, we have two distinct types of policies: horizon-specific policy and cross-insight policy. For these two types of policies, we have designed two corresponding states as inputs, which contain more information than just the price series $\mathcal{P}_t$, as described in Section IV-B1.
- Action a: an action is specified by a portfolio vector $a_t = \{a_{t,1}, a_{t,2}, ..., a_{t,m}\}^T \in \mathrm{R}^m$, where $a_{t,i} \geq 0$ represents the proportion of wealth regarding asset i at time step t, and $\sum_{i=0}^{m} a_{t,i} = 1$.
- Reward r: a reward received from the environment (i.e., the financial market) is based on an agent's action at the current state. A commonly used one is the log return of the portfolio value: $r_t = \log(\frac{a_t^T x_t}{a_{t-1}^T x_{t-1}})$, where x_t is the closing price growth ratio defined as $\frac{p_t}{p_{t-1}} - 1$ and $a_t^T x_t$ represents the accumulative return ratio at time step t.
- Policy π: a policy is essentially a network that outputs a probability distribution (e.g., Gaussian distribution) over actions given the current state s as the input.

IV. Method

According to the fractal market hypothesis (FMH) [9], the stock market price is non-stationary due to the trading among different horizon investors, which makes the portfolio management intractable. Motivated by this intuition, we propose a two-step RL-based approach for portfolio management which learns to incorporate diverse trading policies with different investment horizons to alleviate the non-stationary challenge. The framework is illustrated in Figure 1. In the following, we will delve into more details regarding how we split the horizon information for different policies to ensure that they focus on specific investment horizons, as well as the learning process

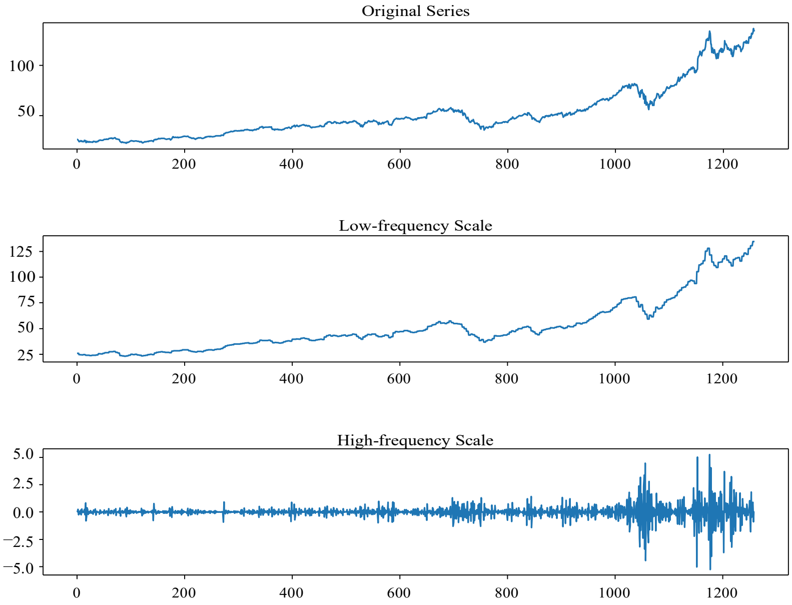


Fig. 2. Visualization of the decomposed horizon information with a granularity of 2, representing long and short-term horizons. The low-frequency scale corresponds to the long-term horizon information, while the high-frequency scale represents the short-term horizon information.

TABLE I
A SUMMARY OF NOTATIONS.

Notation	Description
m	the number of assets
n	the number of horizon-specific policies
P, P^k	the price information of all assets and specific price information decomposed for policy k
$s^k, \tilde{s}$	the state for the k-th horizon-specific policy and cross-insight policy
$a^k, \tilde{a}$	the action taken by the k-th horizon-specific policy and the cross-insight policy
π^k, θ^k	the k-th horizon-specific policy and its parameters
$\tilde{\pi}, \tilde{\theta}$	the cross-insight policy and its parameters
A^k, B^k	the advantage and counterfactual baseline for horizon-specific policy k

within the actor-critic paradigm. We summarize the frequently used notations in Table I.

A. Discrete Wavelet Transform

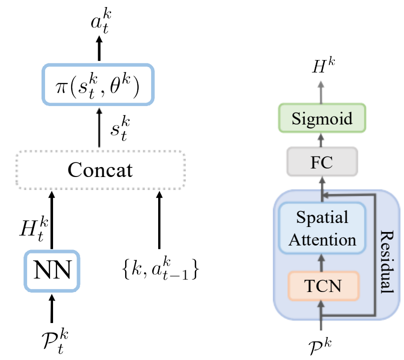
This section introduces how to generate different horizon information for a policy with its respective trading insight. It is well-known that the financial price series can be regarded as a signal with multiple frequencies. The low-frequency component characterizes the coarse structure and the long-term trends of the time series, while the high-frequency component reflects short-term fluctuations. To create different horizon information, we employ a decomposition technique to split the original price series into distinct sub-series based on their frequency bands. For example, we decompose the sub-series containing low-frequency components, which are relevant to long-term investors, and the sub-series containing high-frequency components, which are relevant to short-term investors. This decomposition can be carried out at various levels of granularity. By doing so, each policy can focus on a specific frequency band that aligns with its investment horizon, allowing the policies to gain insights tailored to their respective trading perspectives and make informed decisions.

A few methods are available to decompose price series, such as short-time Fourier transform [31] and wavelet transform [32]. Our work applies the discrete wavelet transform (DWT) method for its excellent performance. DWT acts like a band-pass filter that decomposes a given signal $x(t)$ into multi-level frequency bands. Specifically, in the first level decomposition, the signal $x(t)$ is decomposed into approximation coefficients α^1 and detail coefficients d^1 by passing through a low-pass filter (LPF) and high-pass filter (HPF). The approximation coefficients α^1 maintain the long-term horizon information and the detail coefficients d^1 are associated with the short-term horizon information of the original signal. The decomposition can be formulated as:

$$\alpha^1 = \int x(t)\varphi(t)dt,\ d^1 = \int x(t)\psi(t)dt, \tag{1}$$

and $\varphi()$ and $\psi()$ denote LPF and HPF, respectively [33], where the superscripts of α and d indicate the level of DWT. The previous level output of approximation coefficients will be decomposed in the next iteration until a specified level is reached. For instance, the first level output α^1 would be decomposed into α^2 and d^2. In our framework, we adopt the Haar wavelet transform because of its widespread usage in financial time series processing [34].

After obtaining the coefficients, we mask the coefficients of the frequency band that we want to remove, and then inversely transform the remaining coefficients into the time signal $x^J(t)$. For example, considering the decomposition level is 1, when using DWT to process $P_t = \{P_{t-z}, \ldots, P_{t-1}\} \in \mathrm{R}^{z\times m\times d}$, we can obtain the approximation coefficients $\alpha^1 \in \mathrm{R}^{z/2\times m\times d}$



(a) Actor architecture.

(b) Neural network architecture of the actor.

Fig. 3. Architectures of the actor and its neural network structure.

and detail coefficients $d^1 \in \mathbb{R}^{z/2\times m\times d}$. Masking α^1 and transforming inversely with d^1, we can obtain the high-frequency scale, and vice versa, we can obtain the low-frequency scale. The number of decomposed scales P_t^k is equal to the number of policies, denoted as $P_t^k, k \in \{1, 2, ..., n\}$. The decomposition result is illustrated in Figure 2. We conduct experiments in Section V-C to analyze the influence of different decomposition granularity.

B. Policy Learning

We employ the actor-critic paradigm in our approach, where each actor possesses its own policy, and a centralized critic evaluates the actions of the policies. To ensure diverse and expert performance of the actors within their respective horizons, we address two key challenges: (i) how to learn diversified policies and (ii) how to effectively model the temporal dynamics and asset relationships when extracting features.

1) Actor: Consider that we have a set of n horizon-specific policies parameterized by $\theta = \{\theta^1, ...\theta^n\}$, denoted as $\pi = \{\pi^1, ...\pi^n\}$. To address the challenge (i), each policy $k \in \{1, ..., n\}$ makes pre-decision according to state s^k. The state s^k contains the investment horizon feature H^k extracted from P^k using the well-designed neural network (described in Section IV-B2). Along with the horizon feature, the state s^k also includes the actor-specific ID and the action executed at the previous time step, making the actor behave diversely and preventing the huge changes, respectively, as Figure 3(a) shows. We can write the gradient of the expected return for actor k as:

$$\nabla_{\theta^k} J(\theta^k) = \mathbb{E}_{a^k\sim\pi^k}[\nabla_{\theta^k}\log\pi(a^k|s^k)\Psi_k], \tag{2}$$

where Ψ_k indicates the value estimated by the critic for policy k. In practice, there are many alternatives for Ψ, e.g., the expectation over reward [35], the state value [36], the action advantage [37] and the temporal difference error (TD-error) [38], which lead to a variety of actor-critic algorithms [39]. In our work, we introduce a counterfactual mechanism to calculate the action advantage of each policy and enable multiple policies to be trained in a more effective way.

The cross-insight policy is also an actor, but unlike the horizon-specific policies, the cross-insight policy $\tilde{\pi}$, parameterized by $\tilde{\theta}$ aims to make the final trade decision $\tilde{a}$ based on the pre-decisions made by the horizon-specific policies. Therefore, the state $\tilde{s}$ for the cross-insight policy includes the pre-decisions $\{a^1, ..., a^n\}$ made by different horizon-specific policies. Additionally, the state $\tilde{s}$ also incorporates the features extracted from the original price series of all assets using the neural network described in Section IV-B2, which provides information about the overall market conditions.

$$\nabla_{\tilde{\theta}} J(\tilde{\theta}) = \mathbb{E}_{\tilde{a}\sim\tilde{\pi}}[\nabla_{\tilde{\theta}}\log\tilde{\pi}(\tilde{a}|\tilde{s})Q], \tag{3}$$

where Q is the Q-value estimated by the critic network. The inclusion of both the pre-decisions and the original price series of all assets in the state $\tilde{s}$ enables the cross-insight policy to effectively leverage multiple investment insights and capture the dynamics of the market.

2) Network Structure: To tackle challenge (ii), we provide a well-designed neural network structure for the actors to abstract the temporal dynamic and spatial relationships among the assets. This structure is shown in Figure 3(b).

Temporal Convolution Block. As demonstrated in Figure 3(b), we first utilize a temporal convolution network (TCN) block [40] to extract the temporal dynamic of the assets. Compared with recurrent neural networks, TCN has several appealing properties including facilitating parallel computation, alleviating the gradient exploration/vanishing issue, and demonstrating longer effective memory [8], [40]. After conducting TCN operations on P^k, we obtain an output tensor denoted by $\hat{H}^k \in \mathbb{R}^{m\times f\times z}$, where f is the dimension of hidden features.

Spatial Attention Layer. Afterward, we adopt an attention mechanism [41] to handle the spatial relationships among different assets. Given the output vector of TCN, we calculate the spatial attention weight as:

$$S = V_s \cdot sigmoid((\hat{H}^k W_1)W_2(W_3\hat{H}^{kT})^T + b_s), \tag{4}$$

where T represents the transpose operation, $W_1 \in \mathbb{R}^T$, $W_2 \in \mathbb{R}^{F\times T}$, $W_3 \in \mathbb{R}^F$ and $V_s \in \mathbb{R}^{N\times N}$ are parameters to learn, and $b_s \in \mathbb{R}^{N\times N}$ is the bias vector. The matrix $S \in \mathbb{R}^{N\times N}$ is then normalized by rows to represent the correlation among stocks:

$$S_{i,j} = \frac{\exp(S_{i,j})}{\sum_{u=1}^{m}\exp(S_{i,u})}, \forall 1 \le i \le m. \tag{5}$$

We adopt the residual connection [42] to alleviate the vanishing gradient problem in deep learning. To be precise, the final representation abstracted from P^k is denoted by $H^k = S \times \hat{H}^k + P^k$, and it is then concatenated with the agent ID k and the action executed at the previous time step a_{t-1}^k as the final state s_t^k. The state s_t^k is fed into the policy network to be translated to a vector $\hat{a}^k$ with dimension m. Finally, the vector $\hat{a}^k$ is normalized into an action a^k that satisfies the constraint $\sum_{i=0}^{m} a^k = 1$.

3) Critic: Training the critic network $f_t(\theta^c)$ is another key component of actor-critic RL. The critic parameters θ^c are updated to minimize the loss function as follows:

$$L(\theta^c)=(y^{(\lambda)}-f_t(\mathbf{x};\theta^c))^2, y^{(\lambda)}=(1-\lambda)\sum_{n^*=1}^{\infty}\lambda^{n^*-1}G_t^{(n^*)}, \tag{6}$$

where $y^{(\lambda)}$ is a mixture of n-step returns and $\mathbf{x}$ is the input of the critic network. Note that to distinguish the symbol n representing the number of policies, we use n^* for n-step returns. The calculation of $G_t^{(n^*)}$ can be defined as Eq. (7):

$$G_t^{(n^*)} = \sum_{l=1}^{n^*} {}^{l\ 1}{}_{t+l} + \gamma^{n^*} f_{t+n^*}(\mathbf{x};\theta^c). \tag{7}$$

The critic network is implemented as a two-layer fully-connected network. The input vector $\mathbf{x}$ consists of various components, including the original price series of all assets $\{P_t^1, P_t^2, \ldots, P_t^m\}$ representing the overall market state, the horizon information received by all policies, the trade action $\tilde{a}$ taken by cross-insight policy, and the IDs $\{1, 2, \ldots, n\}$ of all policies. By incorporating these features as inputs to the critic network, we can obtain more accurate evaluations of the actions, which provide better guidance for training all policies.

However, optimizing all policies solely based on the same value estimated by the critic network overlooks the individual contributions of each policy to the cumulative return. To address this issue, we introduce a counterfactual mechanism into the critic network to assess the advantage of each policy. This mechanism enables us to evaluate the unique advantage of each policy and enhance the training effectiveness of the policies.

C. Counterfactual Mechanism

When measuring a policy's contribution to the profit, we may ask what would happen without this policy's action (or the policy's action is replaced with a default action), while the actions of the other policies remain the same. The potential response inferred in hindsight is called a counterfactual response, where the concept *counterfactual* describes the posterior process of reasoning the outcome of alternative actions.

We employ the counterfactual mechanism to evaluate each policy's action. In our scenario, the most straightforward way to evaluate the contribution of a specific policy is to calculate the advantage between the estimated accumulative return after removing the policy's action and the estimated accumulative return without removing the policy's action. In our framework, the accumulative return is represented by the state-action value denoted as $Q()$, which is estimated by the critic network. Considering that the final trade action is the cooperative result of all policies, directly removing the policy's action is not the optimal way to learn a cooperative policy for all policies. Therefore, we choose another alternative, that is, replacing the action of that policy with a default action and then calculating the estimated accumulative return, also known as counterfactual baseline, denoted as B^k. Therefore, the counterfactual mechanism can be described as follows: (1) the centralized critic estimates $Q(\mathbf{x}, \tilde{a}_t)$ for the trade action $\tilde{a}_t$ conditioned on the input $\mathbf{x}$, (2) we compute an advantage value $\hat{A}^k_t$ (also known as credit value) for policy k by comparing $Q(\mathbf{x}, \tilde{a}_t)$ with a counterfactual baseline B^k that marginalizes out the current action of policy k, i.e., replacing its current action with the default action. In other words, $\hat{A}^k_t$ represents the advantage of the policy's current action (the proportion of wealth on the asset) over the default action. $\hat{A}^k > 0$ means the current action is more profitable than the default action of the policy k.

However, how to calculate B^k is a challenging problem and has become a topic for researchers. Many methods have been proposed to calculate the baseline B^k, e.g., using a baseline simulator [43] or calculating the expectation of all possible action-values [44]. However, the above solutions focus on the discrete action space while PM is a task with continuous action space. Generally, for a continuous action space task, the action is a sample with a certain probability from a Gaussian distribution where the mean and standard deviation are learned based on the state/observation. Considering that it is impossible to enumerate all actions in a continuous space, we take an alternative approach by replacing the action of policy k with the Gaussian mean μ^k (learned by the policy network) to calculate the baseline B^k. Choosing μ^k as the default action for calculating B^k can optimize the Gaussian distribution to sample advantaged action with higher probability. If $\hat{A}^k > 0$, the currently selected action is encouraged, and μ^k of the Gaussian distribution would move to the region near to the selected action in the next update (because the region where μ^k is located enjoys a higher probability of being sampled). If $\hat{A}^k < 0$, it indicates that the current action is not encouraged, and μ^k of the Gaussian distribution will move away from the region near to the selected action in the next update, minimizing the probability of this action being sampled. The mathematical representation can be formulated as follows:

$$\begin{aligned}\hat{A}^k &= Q(\mathbf{x}, \tilde{a}_t) - \mathsf{B}^k \\ &= Q(\mathbf{x}, \tilde{a}_t) - Q(\mathbf{x}, (a_t^{-k}, \mu_t^k)),\end{aligned} \tag{8}$$

where a^{-k} represents the trade action that excludes the action of policy k. In this way, we can utilize the centralized critic to analyze scenarios where only policy k's action changes, thereby calculating a distinct counterfactual baseline for each horizon-specific policy. Next, we show that the counterfactual mechanism does not affect the optimal convergence.

Theorem 1. *For a multi-policy actor-critic algorithm with a common critic, we can represent the gradient of the actor as follows:*

$$g = \mathrm{E}_\pi\left[\sum_{k=1}^{n} \nabla_{\theta^k}\log\pi^k\left(a^k|s^k\right)\hat{A}^k\right]. \tag{9}$$

At each iteration K,

$$\liminf_{\mathrm{K}\to\infty} \|\nabla J\| = 0 \quad w.p.1, \tag{10}$$

where J is the reward function, and θ^k represents the parameters of policy k.

More details about the proof can be found in Appendix.

TABLE II
STATISTICS OF DATASETS.

Datasets	Num. of assets	Training day	Testing day
U.S. market	80	2009-01 to 2020-06	2020-07 to 2022-12
H.K. market	45	2009-01 to 2020-06	2020-07 to 2021-07
China market	34	2009-01 to 2020-06	2020-07 to 2021-07

V. EXPERIMENTAL RESULTS

First, we evaluate the profitability of our proposed framework on real-world stock markets to assess its effectiveness. Additionally, we conduct a series of experiments to delve deeper into the performance of the diverse trading policies. These experiments allow us to assess the effectiveness of different decomposition levels and analyze how they impact the overall performance of our framework. Furthermore, we investigate the effect of the counterfactual mechanism, enabling us to understand its contribution to enhance the training effectiveness of the policies.

A. Experimental Settings

In this section, we describe the dataset statistics, baselines, evaluation metrics, and the implementation details of our framework.

Datasets. The datasets we use are derived from real-world markets, providing a comprehensive evaluation of our framework. We obtained the stock data from constituent stocks in the U.S. market, Hong Kong stock exchange (referred to as H.K. market), and Shanghai stock exchange (referred to as China market). The U.S. market dataset includes a bear market in 2020, which allows us to test the effectiveness of our framework in challenging market conditions. The stock data is collected from Yahoo Finance[1]. The statistics of the data are summarized in Table II.

Baselines. Our method is compared with two kinds of advanced methods. (1) Online learning-based methods: OLMAR [4], CRP [18], ONS [2], UP [19] and EG [15]. We run the above models from an open-source project[2]. (2) Deep reinforcement learning-based methods: EIIE [26], A2C [45], DDPG [46] and PPO [47]. We run these models from an open-source library FinRL [30]. The DeepTrader [8] and SARL [6] models are from an open-source trading library TraderMaster[3]. The models used in the cryptocurrency trading [28] or without code/data published [27], [28] are not included in our comparison.

Metrics. Following [27], we evaluate the performance by three metrics. The first metric is the accumulative return (AR):

$$AR = \prod_{t=1}^{T} \left(\frac{P_{t+1,close} - P_{t,close}}{P_{t,close}} \right) \cdot a_t . \qquad (11)$$

The second metric we use is sharp ratio (SR): $SR = \frac{\mathrm{E}(r_t)}{\sigma(r_t)}$, where r_t is the daily return ratio and $\mathrm{E}(r_t)$, $\sigma(r_t)$ represent expectation of r_t and stand deviation of r_t, respectively. SR makes up for the shortcomings of AR that it does not consider risk factors. However, AR and SR ignore the risk of price slumps, which is also essential during investing. The third metric we use to highlight the influence of downward deviations is Calmar ratio (CR). $CR = \frac{S_n}{MDD}$, where S_n is the annualized returns, and MDD denotes the biggest loss from a peak (at time t) to a trough (at time s) and is calculated via

$$MDD = max_{t:s>t} \frac{S_t - S_G}{S_t} .$$

Implementation details. We implement our framework via PyTorch. We adopt Adam optimizer on the training process with a single NVIDIA 3090 GPU. The training step is 50000 for all three datasets, where the batch size is 128. We set the learning rate to 10^{-4} and the weight decay of $l2$ regularizer to 10^{-4}. The window size is set to 252, while the length of n-step return is 5. In addition, the portfolio vector of all policies is initialized by the average assignment. Results are averaged over 5 runs with random initialization seeds for all RL-based methods. The code and data to reproduce our experiments are available[4].

B. Evaluation on Portfolio Management

The performance comparison of the proposed framework with baselines is reported in Table III. According to Table III, we can have the following observations: (1) Overall, our proposed framework demonstrates superior performance compared with online learning-based and deep RL-based methods across all datasets. Specifically, our framework outperforms ONS and CRP by 51% and 87% in terms of SR and CR on the U.S. market dataset. Furthermore, our framework surpasses all RL-based methods, including the state-of-the-art model DeepTrader, by 27% in terms of SR on the U.S. market. These results indicate that our approach, which incorporates and models information from different horizons separately, is more effective than modeling all information together. By capturing and utilizing the distinct patterns and insights associated with different horizons, our framework achieves enhanced performance in portfolio management. This highlights the advantage of our solution in addressing the non-stationary nature of financial markets and leveraging multiple investment horizons. (2) Compared with RL-based methods, online learning-based methods fail to perform well in terms of SR and CR. Some models even turn into an investment loss, e.g., OLMAR. The reason is that these models rely heavily on the handcraft features, resulting in poor generalization ability in the volatile and complex financial environment. By contrast, RL is a more suitable candidate for such a volatile environment because RL can directly output the trading decisions by the trial-and-error interaction with the financial environment to maximize the investment return. This observation strengthens the effectiveness and significance of applying reinforcement learning in the portfolio management task.

Figure 4 shows the cumulative return of compared models vs. trading days in the three real-world stock markets during the backtesting (OLMAR is discarded due to its poor performance). To verify the effectiveness of these models, we also plot the index of the three markets in Figure 4,

[1] https://finance.yahoo.com/

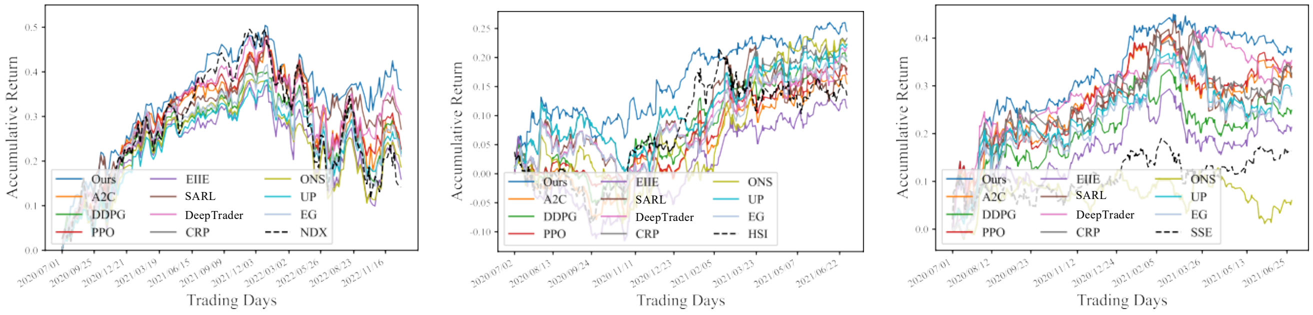
[2] https://github.com/ZhengyaoJiang/PGPortfolio

[3] https://github.com/TradeMaster-NTU/TradeMaster

[4] https://github.com/zhengzetao/Cross-insight-trader

TABLE III
PERFORMANCE COMPARISON OF OUR FRAMEWORK AND OTHER BASELINES.

Categories	Models	U.S. market			H.K. market			China market		
		AR	SR	CR	AR	SR	CR	AR	SR	CR
Online learning-based	OLMAR	-0.52	-1.17	-0.65	-0.65	-2.19	-0.95	-0.57	-0.15	-0.83
	CRP	0.16	0.05	0.21	0.20	1.16	1.60	0.32	1.35	1.77
	ONS	0.15	0.28	0.11	0.21	1.15	1.44	-0.02	-0.07	-0.17
	UP	0.18	0.25	0.13	0.20	1.17	1.62	0.27	1.30	1.98
	EG	0.17	0.14	0.27	0.19	1.16	1.60	0.27	1.29	1.96
Deep RL-based	EIIE	0.13	0.13	0.27	0.13	0.92	1.17	0.19	1.00	1.52
	A2C	0.18	0.15	0.24	0.17	1.22	1.51	0.28	1.38	2.16
	DDPG	0.19	0.33	0.32	0.20	1.31	2.00	0.21	1.13	1.61
	PPO	0.20	0.38	0.30	0.18	1.22	1.85	0.30	1.43	2.34
	SARL	0.22	0.41	0.34	0.18	1.27	1.87	0.28	1.40	2.35
	DeepTrader	0.26	0.47	0.38	0.20	1.30	1.97	0.31	1.44	2.38
	Ours	**0.31**	**0.60**	**0.45**	**0.22**	**1.43**	**2.33**	**0.33**	**1.56**	**2.70**
	Market	0.12	0.34	0.17	0.15	1.01	1.56	0.18	1.16	2.27



(a) U.S. market.

(b) H.K. market.

(c) China market.

Fig. 4. Accumulative return on three datasets during the test period.

i.e., SSE composite index (SSE) for Shanghai stock exchange market, Hangseng index (HSI) for Hong Kong stock exchange and Dow Jones industrial average (DJIA) for U.S. market. From Figure 4, it is evident that our framework consistently outperforms both the market performance (index performance) and other models, regardless of the bull or bear market. This robustness is particularly noteworthy in the U.S. market,where the testing period after the year 2022 experiences a bear market. Despite the challenging market conditions, our cross-insight trader not only outperforms the market but also generates significant profits compared with other models. These results demonstrate the robustness of our approach in different market conditions. Whether the market is experiencing bullish or bearish trends, cross-insight trader consistently achieves superior performance.

C. Ablation Study

1) Analysis of Diverse Policies: We present the trading policies trained using the learning objective outlined in Section IV-B in order to address the following questions: (1) Does the utilization of diverse horizon-specific policies enhance portfolio management performance? (2) What is the optimal number of horizon-specific policies required to achieve the best performance? (3) Do the learned policies exhibit distinct trading styles?

To answer the first question, we discard the horizon-specific policies and only the cross-insight policy remains with the price series of all assets P_t as input, optimized with the Q-value. In this way, our approach degenerates into a traditional A2C algorithm. The experiment result of A2C is reported in Table IV. From Table IV, we can see that the performance of A2C is notably inferior than our approach with horizon-specific policies. This suggests that the inclusion of diverse investment pre-decisions from the horizon-specific policies is indeed effective in improving portfolio management performance. Therefore, we can conclude that the utilization of multiple horizon-specific policies enhances the overall performance compared with relying solely on a single cross-insight policy.

Regarding the second question, we deploy varying numbers of horizon-specific policies and evaluate their individual performances. The corresponding results are presented in Table IV. We notice that as we increase the number of policies deployed (resulting in higher decomposition granularity), the performance of the framework improves. This indicates that having a greater diversity of horizon-specific policies leads to better portfolio management outcomes. By leveraging multiple policies with different investment horizons, our approach benefits from their collective insights and adaptability to various market conditions, resulting in enhanced performance.

TABLE IV
PERFORMANCE COMPARISON OF DIFFERENT NUMBERS OF HORIZON-SPECIFIC POLICIES.

Models	U.S. market			H.K. market			China market		
	AR	SR	CR	AR	SR	CR	AR	SR	CR
A2C	0.18	0.15	0.24	0.17	1.22	1.51	0.28	1.38	2.16
2 policies	0.21	0.23	0.29	0.20	1.27	1.73	0.29	1.41	2.40
3 policies	0.25	0.33	0.33	0.20	1.28	1.96	0.30	1.45	2.48
4 policies	0.29	0.45	0.40	0.21	1.30	2.05	0.31	1.47	2.53
5 policies	**0.31**	**0.60**	**0.45**	**0.22**	**1.43**	**2.33**	**0.33**	**1.56**	**2.70**

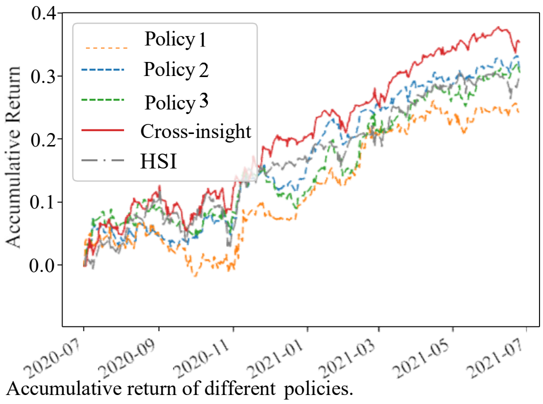


Fig. 5. Accumulative return of different policies.

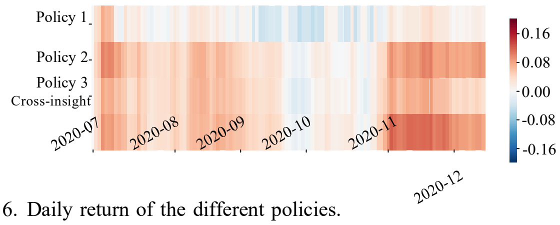


Fig. 6. Daily return of the different policies.

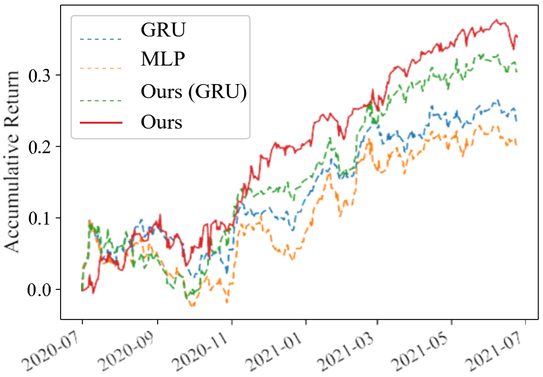


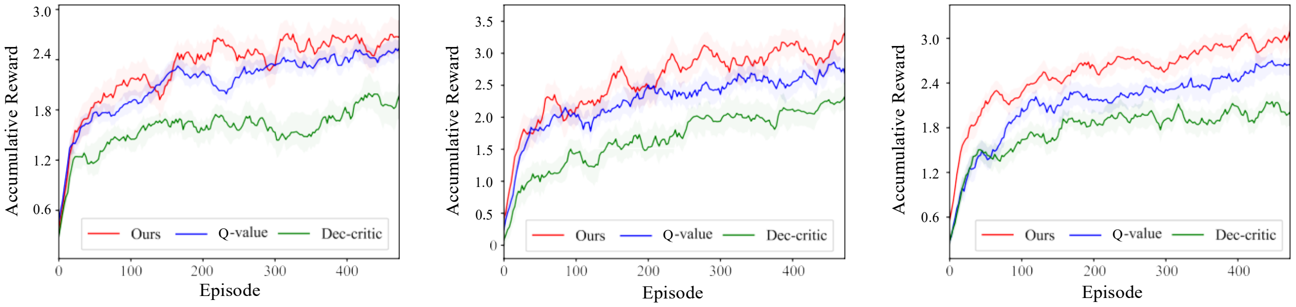
Fig. 7. Accumulative return of the actor with different neural network structures.

To answer the third question, we visualize the cumulative return of each policy in Figure 5 and plot the daily return of each policy in Figure 6. We conduct this experiment on the H.K. market, where policies 1, 2, and 3 are responsible for the short, middle, and long-term investment horizons, respectively. As observed, the wealth generated by the fused policy significantly outperforms that of the market index (HSI) and all other individual policies. However, the policy focusing on the short-term horizon exhibits inferior performance compared with the others. This discrepancy can be attributed to the limited amount of information available for short-term predictions, making it more challenging to achieve consistent profits. Furthermore, Figure 6 demonstrates the diversified performance of the three policies. The middle-term and long-term investment horizons exhibit more stable returns, while the short-term investment horizon shows greater volatility (evident from the color transition). This diversification indicates that each policy leverages its specific insight and adapts to its respective investment horizon, resulting in varied performance characteristics. Overall, the visualizations illustrate the effectiveness of our framework, particularly the cross-insight policy, in outperforming the market index and utilizing diverse trading strategies across different investment horizons.

2) Analysis of Neural Network: To investigate the effectiveness of the well-designed neural network in the actor component, we compare our proposed approach with three variants. The first two variants, referred to as MLP and GRU, replace the well-designed neural network with a multilayer perceptron (MLP) and a gated recurrent unit (GRU), respectively. The third variant, denoted as ours (GRU), retains the same network structure as our approach but replaces the temporal convolutional network (TCN) with a GRU unit. The results are presented in Figure 7.

The performance of our cross-insight Trader, which incorporates spatial attention, surpasses all three variant methods. This empirical finding confirms the critical importance of effectively considering the correlations among assets (i.e., in methods ours and ours (GRU)) when extracting temporal patterns. Furthermore, using TCN for temporal pattern extraction proves to be more effective than employing the GRU unit. On the other hand, the methods that solely focus on the temporal patterns without considering asset correlations (i.e., GRU and MLP) yield inferior performance. This comparison reinforces the significance of our well-designed neural network and its ability to capture both temporal and spatial information, leading to superior performance in portfolio management tasks.

3) Analysis of Counterfactual Mechanism: To evaluate the effectiveness of our counterfactual mechanism, we compare it with two variants. The first variant optimizes all policies with the same Q-value, while our counterfactual mechanism assigns different values to each policy based on their contribution to



(a) U.S. market. (b) H.K. market. (c) China market.

Fig. 8. Learning curves for counterfactual policy and other two variants on three datasets during training.

the accumulative return. The second variant, called Dec-critic, adopts a decentralized critic structure, where each actor has its own critic, and each critic only receives the corresponding policy's state and executed action at the previous time step as input. The learning curves, as shown in Figure 8, clearly demonstrate that our counterfactual mechanism outperforms the variant that optimizes all policies with the same value. This achievement can be attributed to the different advantages provided by our counterfactual mechanism. By considering the individual contribution of each policy, the policies are guided to train more effectively, leading to improved performance in portfolio management. On the other hand, the performance of the Dec-critic variant is inferior compared with the other variants. This is due to that each agent has its own critic and is trained independently, without a collaborative effort to maximize profits. This result confirms that using a centralized critic and incorporating a counterfactual mechanism to evaluate each policy's advantage is a superior approach for our framework. Overall, these findings highlight the effectiveness of our counterfactual mechanism in guiding the policies' behavior and improving the performance of our framework in portfolio management.

VI. Conclusion

This paper presents cross-insight trader, a two-step reinforcement learning-based approach designed to adapt to changing market conditions by integrating multiple trading policies with different investment horizons. Our approach leverages tailored horizon information to learn horizon-specific policies in the first step. In the second step, the approach learns a cross-insight policy that considers the investment pre-decisions made by the horizon-specific policies to make the final trade decision. The cross-insight policy is optimized based on the Q-value, while the horizon-specific policies are optimized using advantage values derived from a counterfactual mechanism that measures each policy's contribution to the return. Experimental results on three markets show the superior performance of cross-insight trader compared with existing baselines.

Further analysis validates the effectiveness of incorporating different investment horizons into the decision-making process. However, it is worth noting that the approach's performance is currently limited by using only the closing price as input for prediction. To enhance the approach's performance, future work could explore the incorporation of additional sources of information into the state representation.

Appendix

Proof. The gradient of the counterfactual baseline is given by:

$$g = \mathbb{E}_\pi\left[\sum_{k=1}^{n} \nabla_{\theta^k} \log \pi^k\left(a^k \mid s^k\right) \hat{A}^k\right], \tag{12}$$

where $\hat{A}^k$ is expressed as

$$\hat{A}^k = Q(\mathbf{x}, \tilde{a}) - \mathrm{B}^k, \tag{13}$$

in which B^k is the counterfactual baseline defined in Eq. (8). Therefore, Eq. (12) can be represented as:

$$\begin{aligned} g &= g_1 + g_2 \\ &= \mathbb{E}_\pi\left[\sum_{k=1}^{n} \nabla_{\theta^k} \log \pi^k\left(a^k \mid s^k\right) Q(\mathbf{x}, \tilde{a})\right] \\ &\quad - \mathbb{E}_\pi\left[\sum_{k=1}^{n} \nabla_{\theta^k} \log \pi^k\left(a^k \mid s^k\right) \mathrm{B}^k\right]. \end{aligned} \tag{14}$$

First, we consider g_2:

$$g_2 = -\mathbb{E}_\pi\left[\sum_{k=1}^{n} \nabla_{\theta^k} \log \pi^k\left(a^k \mid s^k\right) \mathrm{B}^k\right]. \tag{15}$$

Let d_π be the discounted state distribution as defined by [45]:

$$\begin{aligned} g_2 &= -\int_s d_\pi(s) \sum_{k=1}^{n} \int_{a^{-k}} \pi^{-k}\left(a^{-k}|s^{-k}\right) \cdot \int_{a^k} \pi^k\left(a^k|s^k\right) \nabla_{\theta^k} \log \pi^k\left(a^k|s^k\right) \mathrm{B}^k \\ &= -\int_s d^\pi(s) \sum_{k=1}^{n} \int_{a^{-k}} \pi^{-k}\left(a^{-k}|s^{-k}\right) \cdot \int_{a^k} \nabla_{\theta^k} \pi^k\left(a^k|s\right) \mathrm{B}^k \\ &= -\int_s d^\pi(s) \sum_{k=1}^{n} \int_{a^{-k}} \pi^{-k}\left(a^{-k}|s^{-k}\right) \mathrm{B}^k \cdot \nabla_{\theta^k} \int_{a^k} \pi^k\left(a^k|s^k\right) \\ &= -\int_s d^\pi(s) \sum_{k=1}^{n} \int_{a^{-k}} \pi^{-k}\left(a^{-k}|s^{-k}\right) \mathrm{B}^k \nabla_\theta 1 \\ &= 0 \end{aligned} \tag{16}$$

It is clearly that the baseline B^k reduces variance but does not change the expected gradient, and therefore the counterfactual baseline does not affect the convergence of the multi-agent reinforcement learning.

The remainder of the expected policy gradient g_1 is given by:

$$g_1 = \mathrm{E}_\pi \left[\sum_{k=1}^{n} \nabla_{\theta^k} \log \pi^k\left(a^k|s^k\right) Q(\mathbf{x}, \tilde{a}) \right]. \tag{17}$$

It is proved in [48] that an actor-critic following this gradient converges to a local maximum of the expected reward. □

REFERENCES

[1] B. Li and S. C. H. Hoi, "Online portfolio selection: A survey," *ACM Comput. Surv.*, vol. 46, no. 3, pp. 35:1–35:36, 2014.

[2] A. Agarwal, E. Hazan, S. Kale, and R. E. Schapire, "Algorithms for portfolio management based on the newton method," in *Machine Learning, Proceedings of the Twenty-Third International Conference, ICML 2006*, 2006, pp. 9–16.

[3] A. Borodin, R. El-Yaniv, and V. Gogan, "Can we learn to beat the best stock," in *Advances in Neural Information Processing Systems 16 Neural Information Processing Systems, NIPS 2003*, 2003, pp. 345–352.

[4] B. Li and S. C. H. Hoi, "On-line portfolio selection with moving average reversion," in *Proceedings of the 29th International Conference on Machine Learning, ICML 2012*, 2012.

[5] C. J. Neely, D. E. Rapach, J. Tu, and G. Zhou, "Forecasting the equity risk premium: The role of technical indicators," *Manag. Sci.*, vol. 60, no. 7, pp. 1772–1791, 2014.

[6] Y. Ye, H. Pei, B. Wang, P. Chen, Y. Zhu, J. Xiao, and B. Li, "Reinforcement-learning based portfolio management with augmented asset movement prediction states," in *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020*, 2020, pp. 1112–1119.

[7] H. Niu, S. Li, and J. Li, "Metatrader: An reinforcement learning approach integrating diverse policies for portfolio optimization," in *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, 2022, pp. 1573–1583.

[8] Z. Wang, B. Huang, S. Tu, K. Zhang, and L. Xu, "Deeptrader: A deep reinforcement learning approach for risk-return balanced portfolio management with market conditions embedding," in *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021*, 2021, pp. 643–650.

[9] L. Kristoufek, "Fractal markets hypothesis and the global financial crisis: Wavelet power evidence," *Sci. Rep.*, vol. 3, p. 2857, 2013.

[10] E. E. Peters, *Fractal Market Analysis: Applying Chaos Theory to Investment and Economics*. Wiley, 1994.

[11] Z. Li and V. W. L. Tam, "Combining the real-time wavelet denoising and long-short-term-memory neural network for predicting stock indexes," in *2017 IEEE Symposium Series on Computational Intelligence, SSCI 2017*, 2017, pp. 1–8.

[12] P. Liu, Y. Zhang, F. Bao, X. Yao, and C. Zhang, "Multi-type data fusion framework based on deep reinforcement learning for algorithmic trading," *Appl. Intell.*, vol. 53, no. 2, pp. 1683–1706, 2023.

[13] X. Wu, H. Chen, J. Wang, L. Troiano, V. Loia, and H. Fujita, "Adaptive stock trading strategies with deep reinforcement learning methods," *Inf. Sci.*, vol. 538, pp. 142–158, 2020.

[14] J. Lee, R. Kim, S. Yi, and J. Kang, "MAPS: multi-agent reinforcement learning-based portfolio management system," in *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*, 2020, pp. 4520–4526.

[15] D. P. Helmbold, R. E. Schapire, Y. Singer, and M. K. Warmuth, "On-line portfolio selection using multiplicative updates," *Math. Financ.*, vol. 8, no. 4, pp. 325–347, 1998.

[16] Y. Li, X. Zheng, C. Chen, J. Wang, and S. Xu, "Exponential gradient with momentum for online portfolio selection," *Expert Syst. Appl.*, vol. 187, p. 115889, 2022.

[17] A. Borodin, R. El-Yaniv, and V. Gogan, "On the competitive theory and practice of portfolio selection (extended abstract)," in *LATIN 2000: Theoretical Informatics, 4th Latin American Symposium, Proceedings*, 2000, pp. 173–196.

[18] T. M. Cover and D. H. Gluss, "On-line portfolio selection using multiplicative updates," *Adv. Appl. Math.*, vol. 7, no. 2, pp. 170–181, 1986.

[19] A. Blum and A. Kalai, "Universal portfolios with and without transaction costs," *Mach. Learn.*, vol. 35, no. 3, pp. 193–205, 1999.

[20] B. Li, P. Zhao, S. C. H. Hoi, and V. Gopalkrishnan, "PAMR: passive aggressive mean reversion strategy for portfolio selection," *Mach. Learn.*, vol. 87, no. 2, pp. 221–258, 2012.

[21] B. Li, S. C. H. Hoi, P. Zhao, and V. Gopalkrishnan, "Confidence weighted mean reversion strategy for online portfolio selection," *ACM Trans. Knowl. Discov. Data*, vol. 7, no. 1, pp. 4:1–4:38, 2013.

[22] D. Huang, J. Zhou, B. Li, S. C. H. Hoi, and S. Zhou, "Robust median reversion strategy for on-line portfolio selection," in *IJCAI2013, Proceedings of the 23rd International Joint Conference on Artificial Intelligence*, 2013, pp. 2006–2012.

[23] R. Neuneier, "Optimal asset allocation using adaptive dynamic programming," in *Advances in Neural Information Processing Systems 8, NIPS 1995*, 1995, pp. 952–958.

[24] A. Tsantekidis, N. Passalis, A. Toufa, K. S. Zarkias, S. Chairistanidis, and A. Tefas, "Price trailing for financial trading using deep reinforcement learning," *IEEE Trans. Neural Networks Learn. Syst.*, vol. 32, no. 7, pp. 2837–2846, 2021.

[25] J. Jang and N. Seong, "Deep reinforcement learning for stock portfolio optimization by connecting with modern portfolio theory," *Expert Syst. Appl.*, vol. 218, p. 119556, 2023.

[26] Z. Jiang, D. Xu, and J. Liang, "A deep reinforcement learning framework for the financial portfolio management problem," *CoRR*, vol. abs/1706.10059, 2017.

[27] K. Xu, Y. Zhang, D. Ye, P. Zhao, and M. Tan, "Relation-aware transformer for portfolio policy learning," in *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*, 2020, pp. 4647–4653.

[28] Y. Zhang, P. Zhao, Q. Wu, B. Li, J. Huang, and M. Tan, "Cost-sensitive portfolio selection via deep reinforcement learning," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 1, pp. 236–248, 2022.

[29] Y. Deng, F. Bao, Y. Kong, Z. Ren, and Q. Dai, "Deep direct reinforcement learning for financial signal representation and trading," *IEEE Trans. Neural Networks Learn. Syst.*, vol. 28, no. 3, pp. 653–664, 2017.

[30] X. Liu, H. Yang, J. Gao, and C. D. Wang, "Finrl: deep reinforcement learning framework to automate trading in quantitative finance," in *ICAIF'21: 2nd ACM International Conference on AI in Finance*, 2021, pp. 1:1–1:9.

[31] K. Samiee, P. Kovács, and M. Gabbouj, "Epileptic seizure classification of EEG time-series using rational discrete short-time fourier transform," *IEEE Trans. Biomed. Eng.*, vol. 62, no. 2, pp. 541–552, 2015.

[32] M. Xu, M. Han, and H. Lin, "Wavelet-denoising multiple echo state networks for multivariate time series prediction," *Inf. Sci.*, vol. 465, pp. 439–458, 2018.

[33] S. Lahmiri, "Wavelet low- and high-frequency components as features for predicting stock prices with backpropagation neural networks," *J. King Saud Univ. Comput. Inf. Sci.*, vol. 26, no. 2, pp. 218–227, 2014.

[34] G. Liu, Y. Mao, Q. Sun, H. Huang, W. Gao, X. Li, J. Shen, R. Li, and X. Wang, "Multi-scale two-way deep neural network for stock trend prediction," in *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*, 2020, pp. 4555–4561.

[35] R. J. Williams, "Simple statistical gradient-following algorithms for connectionist reinforcement learning," *Mach. Learn.*, vol. 8, pp. 229–256, 1992.

[36] C. J. C. H. Watkins and P. Dayan, "Technical note q-learning," *Mach. Learn.*, vol. 8, pp. 279–292, 1992.

[37] L. Weaver and N. Tao, "The optimal reward baseline for gradient-based reinforcement learning," in *UAI '01: Proceedings of the 17thConference in Uncertainty in Artificial Intelligence*, 2001, pp. 538–545.

[38] R. S. Sutton, "Learning to predict by the methods of temporal differences," *Mach. Learn.*, vol. 3, pp. 9–44, 1988.

[39] R. S. Sutton and A. G. Barto, "Reinforcement learning: An introduction," *IEEE Trans. Neural Networks*, vol. 9, no. 5, pp. 1054–1054, 1998.

[40] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," in *4th International Conference on Learning Representations, ICLR 2016, Conference Track Proceedings*, 2016.

[41] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017*, 2017, pp. 5998–6008.

[42] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016*, 2016, pp. 770–778.

[43] D. H. Wolpert and K. Tumer, "Optimal payoff functions for members of collectives," *Adv. Complex Syst.*, vol. 4, no. 2-3, pp. 265–280, 2001.

[44] J. N. Foerster, G. Farquhar, T. Afouras, N. Nardelli, and S. Whiteson, "Counterfactual multi-agent policy gradients," in *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18)*, 2018, pp. 2974–2982.

[45] R. S. Sutton, D. A. McAllester, S. P. Singh, and Y. Mansour, "Policy gradient methods for reinforcement learning with function approximation," in *Advances in Neural Information Processing Systems 12, NIPS 1999*, 1999, pp. 1057–1063.

[46] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," in *4th International Conference on Learning Representations, ICLR 2016*, 2016.

[47] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *CoRR*, vol. abs/1707.06347, 2017.

[48] V. R. Konda and J. N. Tsitsiklis, "Actor-critic algorithms," in *Advances in Neural Information Processing Systems 12, [NIPS Conference 1999]*, 1999, pp. 1008–1014.

Zetao Zheng is currently a Ph.D. student with the Center for Future Media, School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, China. His research interests include graph neural network and time series prediction.

Jie Shao received the B.E. degree in computer science from Southeast University, Nanjing, China, in 2004 and the Ph.D. degree in computer science from The University of Queensland, Brisbane, Australia, in 2009. He is currently a Professor with the Center for Future Media, School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, China. He worked as a Research Fellow at the University of Melbourne from 2008 to 2011, and at National University of Singapore from 2012 to 2014. His research interests include database management and multimedia information retrieval.

Shilong Deng is a master student majoring in Computer Science and Technology at the School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, China. His research interests include reinforcement learning and quantitative finance.

Anjie Zhu received the B.E. degree in University of Electronic Science and Technology of China in 2019. She is currently a Ph.D. student with the Center for Future Media, School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, China. Her research interests include reinforcement learning and multimodal learning.

Feiyu Chen received the M.E. degree in electrical and electronic engineering from the University of Nottingham, UK, in 2016 and the Ph.D. degree in computer science and technology from University of Electronic Science and Technology of China, Chengdu, China in 2022. She is currently a joint postdoctor with University of Electronic Science and Technology of China and Sichuan Artificial Intelligence Research Institute, Yibin, China. Her research interests include natural language processing and multimodal learning.

Heng Tao Shen Heng Tao Shen is the Dean of School of Computer Science and Engineering, the Executive Dean of AI Research Institute at University of Electronic Science and Technology of China (UESTC). He obtained his BSc with 1st class Honours and PhD from Department of Computer Science, National University of Singapore in 2000 and 2004 respectively. His research interests mainly include Multimedia Search, Computer Vision, Artificial Intelligence, and Big Data Management. He is/was an Associate Editor of ACM Transactions of Data Science, IEEE Transactions on Image Processing, IEEE Transactions on Multimedia, IEEE Transactions on Knowledge and Data Engineering, and Pattern Recognition. He is a Fellow of ACM, IEEE and OSA.