IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 17 October 2025, accepted 7 November 2025, date of publication 13 November 2025,
date of current version 19 November 2025.

Digital Object Identifier 10.1109/ACCESS.2025.3632201

RESEARCH ARTICLE

Multi-Asset Multi-Agent Reinforcement Learning for Portfolio Management

SANG-HO KIM AND KI-HOON LEE
School of Computer and Information Engineering, Kwangwoon University, Nowon-gu, Seoul 01897, Republic of Korea

Corresponding author: Ki-Hoon Lee (kihoonlee@kw.ac.kr)

This work was supported in part by the National Research Foundation of Korea (NRF) grant funded by Korean Government (MSIT) under Grant NRF-2022R1F1A1062787; and in part by the Research Grant of Kwangwoon University, in 2023.

ABSTRACT Portfolio management reduces the risks and improves the profits of a portfolio comprising various asset classes (including stocks, bonds, commodities, and cash) that exhibit low correlations and distinct risk-return characteristics. The five key challenges in portfolio management are 1) asset allocation, 2) security selection, 3) security allocation, 4) adaptation to concept drift, and 5) consideration of interest rates. Concept drift denotes a shift in the statistical properties of the data over time. Artificial intelligence-based methods using deep learning or reinforcement learning (RL) have recently been proposed to address these challenges, but they focus only on subsets of the challenges. To address all five key challenges, we propose a multi-asset multi-agent RL framework named MAMA, which integrates asset allocation, security selection, and security allocation, as well as considers both concept drift and interest rates. MAMA comprises three main components: 1) security selection and allocation, 2) asset allocation, and 3) portfolio construction and execution. For security selection and allocation, we propose a multi-agent structure that captures the risk-return characteristics of each asset class. Each intra-asset agent consists of graph neural network-based state representation learning (SRL), which selects the most profitable securities while adapting to concept drift at the security level, and intra-asset RL, which determines their investment ratios. For asset allocation, we integrate non-continual and continual inter-asset agents to adapt to concept drift at the asset class level. Each inter-asset agent consists of recurrent neural network-based SRL, which captures changes in interest rates, and inter-asset RL, which determines investment ratios among asset classes. For portfolio construction and execution, we compute the investment ratios and implement them through a trading system. The performance of MAMA is compared with that of state-of-the-art asset allocation, security selection, and security allocation methods. Experimental results for securities in stocks, bonds (including treasuries), and commodities in the United States reveal that MAMA achieves a compounded annual growth rate of 40.9%, surpassing the second-best method by 23.0%P.

INDEX TERMS Portfolio management, security selection, security allocation, asset allocation, non-continual learning, continual learning, reinforcement learning.

I. INTRODUCTION

A portfolio is a combination of various asset classes, including stocks, bonds, commodities, and cash [1]. Portfolio management aims to mitigate the risks and improve the profits of the portfolio by leveraging the asset classes, which exhibit low correlations and distinct risk-return characteristics [2]. To this end, investors are required to determine investment ratios for the aforementioned asset classes, and then select securities within each asset class and determine their respective investment ratios. Fig. 1 illustrates an example of portfolio management. First, investment ratios of 30%, 30%, 30%, and 10% are determined for four asset classes. Second, three securities are selected within each asset class: AAPL, NVDA, and AMZN for stocks; BIL, VGIT, and TLT for bonds; and GLD, USO, and SLV for

The associate editor coordinating the review of this manuscript and approving it for publication was Wei Wang.

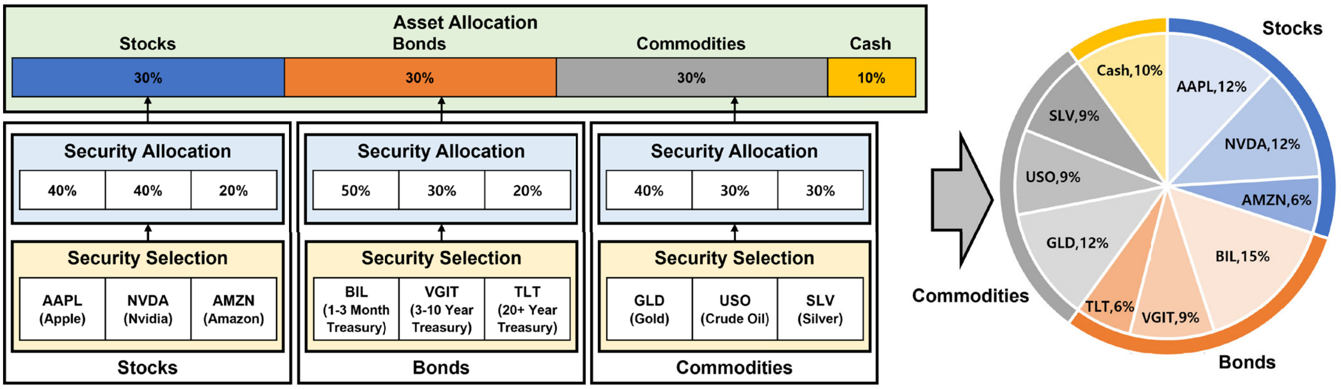


FIGURE 1. Example of portfolio management.

commodities. Third, investment ratios are determined among the selected securities: 40%, 40%, and 20% for AAPL, NVDA, and AMZN, respectively; 50%, 30%, and 20% for BIL, VGIT, and TLT, respectively; and 40%, 30%, and 30% for GLD, USO, and SLV, respectively. Finally, the portfolio is constructed based on the selected securities and cash, along with their respective investment ratios, as shown on the right side of Fig. 1.

Portfolio management faces five prominent challenges. The first is how much to invest in each asset class at the asset class level. The second is how to select profitable securities among numerous candidates within each asset class at the security level. The third is how much to invest in each selected security at the security level. The fourth is how to maintain an optimal portfolio while adapting to concept drift, which manifests differently at the security and asset class levels, caused by the non-stationary nature of financial markets. Concept drift is the phenomenon in which the statistical distribution of data changes over time. The fifth is how to account for the impact of monetary policy at the asset class level. Artificial intelligence (AI) technologies have played a pivotal role across various domains, including finance [3], [4], [5], healthcare [6], and internet of things [7]. Recently, AI-based methods [8], [9], [10], [11], [12], [13], [14], [15], [16], [17], [18], [19], [20], [21], [22], [23], [24], [25], [26], [27], [28], [29], [30], [31], [32], [33], [34], [35], [36], [37], [38], [39] for portfolio management, which use deep learning (DL) or reinforcement learning (RL), have been proposed to accommodate these challenges, but they focus only on subsets of the five challenges. Combining all five challenges into a unified framework is technically difficult because it requires both multi-level decision-making and the ability to adapt to financial market changes. To the best of our knowledge, a comprehensive AI-based method that simultaneously addresses all of these challenges has yet to be proposed, even though it is crucial for achieving fully autonomous and expert-level portfolio management.

Most asset allocation methods [8], [9], [10], [11], [12], [19] adopt static or dynamic strategies on a fixed set of exchange-traded funds (ETFs) covering various asset classes, typically with periodic rebalancing, to determine investment ratios among asset classes. Static asset allocation methods, such as the All-Weather strategy [12], initially define the investment ratios allocated to the ETFs and maintain them without change. Dynamic asset allocation methods, such as the dual momentum strategy [19], adjust the investment ratios allocated to the ETFs adaptively based on market conditions, such as momentum or signals derived from technical indicators [40]. However, the static and dynamic asset allocation methods have three key challenges. First, they overlook the security selection and allocation problems, thereby limiting opportunities to enhance returns and mitigate risks within each asset class. Second, they neglect the impact of monetary policy. In particular, interest rates are widely regarded as the most important factor, as they directly influence the overall movement of financial markets [41], [42]. Third, the application of AI to the asset allocation problem remains relatively underexplored.

For the selection of profitable securities, most existing security selection methods [13], [14], [17], [18], [20], [21], [22], [23], [24], [29], [30], [31], [32] have adopted graph neural networks (GNNs) to capture the relational dependencies between securities. Although these methods show promising performance, they have two key limitations. First, they overlook the asset allocation problem by focusing solely on a single asset class (i.e., stocks). This can result in significant losses as the entire portfolio is directly exposed to the risks inherent in a single asset class. Second, most of them ignore the security allocation problem by investing in each security in equal proportions, which is not optimal and limits their profitability.

To determine the investment ratios among securities, most existing security allocation methods [15], [16], [25], [26], [27], [28], [33], [34], [35], [36], [37], [38], [39] adopt RL or algorithmic strategies for a fixed set of securities. Although these methods demonstrate promising performance, they also have two key limitations. First, they overlook the asset allocation problem. Second, they neglect the security

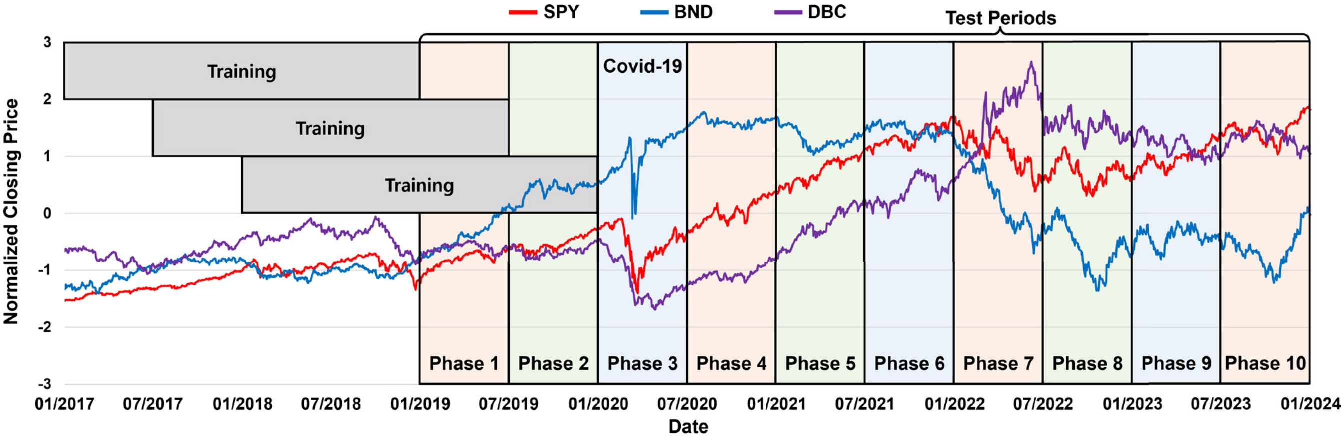


FIGURE 2. Normalized closing prices of SPY, BND, and DBC ETFs.

selection problem in which the set of securities changes owing to concept drift at the security level, including the listing and delisting of securities. Because these security allocation methods do not consider these changes, their ability to adapt to financial markets is limited.

To adapt to concept drift at the security level, the most recent method [32] combines non-continual and continual learning for security selection. Concept drift occurs owing to the varying levels of volatility in financial markets. Abrupt concept drift occurs in high-volatility markets (e.g., phase 3 in Fig. 2), whereas gradual concept drift occurs in low-volatility markets (e.g., phase 2 in Fig. 2). Non-continual learning is preferred for handling abrupt concept drift because it does not consider knowledge retention [43] of information obtained in previous tasks (i.e., phases). However, non-continual learning may not be beneficial in low-volatility markets. In contrast, continual learning is preferred for handling gradual concept drift because it retains knowledge by leveraging the information obtained in previous tasks. However, continual learning may not be beneficial in high-volatility markets. Although this method demonstrates promising performance in security selection, its application at the asset class level (i.e., the asset allocation problem) remains unexplored.

In this study, we propose a multi-asset multi-agent RL framework named MAMA, which addresses all five aforementioned challenges. Our main contribution is a unified framework that not only combines asset allocation, security selection, and security allocation, but also considers both concept drift and interest rates. As illustrated in Fig. 3, our framework comprises three components: (1) security selection and allocation, (2) asset allocation, and (3) portfolio construction and execution. For security selection and allocation, we devise a multi-agent structure that independently learns the risk-return characteristics of each asset class. Each intra-asset agent comprises GNN-based state representation learning (SRL) and intra-asset RL to select profitable securities and determine their respective investment ratios.

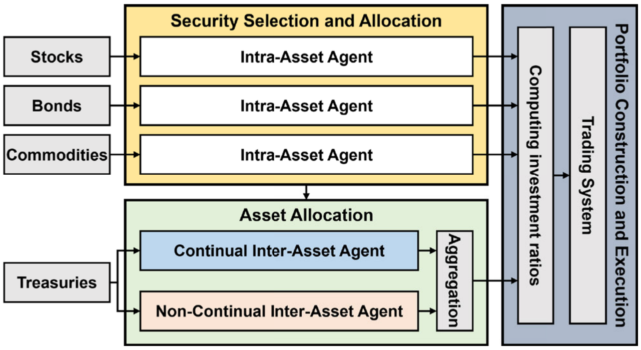

Security Selection and Allocation
Stocks
Bonds
Commodities
Intra-Asset Agent
Intra-Asset Agent
Intra-Asset Agent
Asset Allocation
Treasuries
Continual Inter-Asset Agent
Non-Continual Inter-Asset Agent
Aggregation
Computing investment ratios
Trading System
Portfolio Construction and Execution

FIGURE 3. Our framework.

GNN-based SRL is used to capture both temporal and relational dependencies between securities while adapting to concept drift at the security level. For asset allocation, we combine non-continual and continual inter-asset agents to adapt to concept drift at the asset class level. Each inter-asset agent comprises recurrent neural network (RNN)-based SRL and inter-asset RL to determine the investment ratios among the asset classes. RNN-based SRL is used to learn changes in interest rates by capturing temporal dependencies from treasuries with various maturities, ranging from months to over 20 years. For portfolio construction and execution, we construct the portfolio by computing investment ratios and execute it using a trading system. The proposed MAMA achieves a compounded annual growth rate (CAGR) of 40.9% compared to state-of-the-art asset allocation, security selection, and security allocation methods, outperforming the second-best method [32] by 23.0%P for securities in stocks, bonds (including treasuries with various maturities), and commodities in the United States.

The remainder of this paper is organized as follows. In Section II, we review existing works on related topics. In Sections III and IV, we introduce the proposed MAMA method and present the experimental results, respectively.

Finally, we outline our conclusions and suggestions for future research in Section V.

II. RELATED WORK

Our study builds on recent works on security selection, security allocation, and asset allocation methods in the investment field.

A. SECURITY SELECTION METHODS

Several security selection methods [13], [14], [17], [18], [20], [21], [22], [23], [24], [29], [30], [31], [32] have been proposed to capture inter-security dependencies. Kim et al. [13] presented a hierarchical graph attention network (GAT) that consolidates information obtained from neighboring securities and various relationships. Feng et al. [14] proposed a relational security-ranking framework, which combines temporal graph convolutions to extract time-sensitive relationships between securities. Sawhney et al. [17] introduced a spatio-temporal hypergraph attention network that combines spatial hypergraph convolutions and the Hawkes process with an attention mechanism to learn spatial and temporal dependencies simultaneously. Hsu et al. [18] presented a financial GAT to select securities based on hierarchical relationships between securities and sectors. Kim et al. [20] proposed a portfolio management framework that integrates ranking models, classification, and regression models for selecting securities and determining their investment ratios. Feng et al. [21] employed a relation-aware dynamic attributed GAT to model the topological structure of local correlation. He et al. [22] presented a static-dynamic GNN to capture potential relationships between securities. Wang et al. [23] presented an adaptive long-short pattern transformer to refine the security patterns at various context scales. Ma et al. [24] proposed an attribute-driven fuzzy hypergraph network to measure the intensity of collective relationships and simulate the influence of securities. Huynh et al. [29] proposed a profit-driven framework to capture higher-order correlations and security-specific patterns. Song et al. [30] proposed a multi-relational graph attention ranking network to capture dependencies between securities based on industry, Wiki, and price similarity relationships. Tian et al. [31] proposed a graph evolution-based recurrent unit that captures dynamic temporal dependencies between securities using time-series data. Kim et al. [32] proposed a diversified adaptive security selection framework named DASS, which integrates non-continual and continual graph learning to adapt to various market volatilities. However, these studies did not consider the asset allocation problem, and most of them also did not consider the security allocation problem. In this paper, we determine investment ratios not only at the security level but also at the asset class level.

B. SECURITY ALLOCATION METHODS

Several security allocation methods [15], [16], [25], [26], [27], [28], [33], [34], [35], [36], [37], [38], [39] have been proposed to determine optimal investment ratios corresponding to predetermined sets of securities. Ye et al. [15] proposed a state-augmented RL framework, combining heterogeneous data types, including news and security prices. Lee et al. [16] proposed a multi-agent RL-based portfolio management system to create a diversified portfolio. Lim et al. [25] proposed an RL agent that uses long short-term memory (LSTM) to reduce the time delay of technical indicators by predicting future stock prices. Huang and Tanaka [26] proposed a modularized multi-agent RL-based system that combines evolving and strategic agent modules to enhance the scalability and reusability in financial portfolio management. Shi et al. [27] proposed a graph convolutional network-based RL framework to capture multi-scale temporal and relational dependencies. Lin et al. [28] proposed a multi-agent-based deep RL framework that uses a two-level nested agent structure to account for both microscopic and macroscopic perspectives. Zhang et al. [33] proposed a multi-agent RL method that employs a portfolio insurance strategy to reduce risks. Park et al. [34] presented a risk-sensitive multi-agent network that considers both market and parameter uncertainties. Ma and Nan [35] proposed a dynamic wavelet coherence graph convolutional RL algorithm to capture dynamic time-frequency dependencies with respect to different periods. Li et al. [36] developed a multi-agent and self-adaptive framework named MASA, which combines RL- and solver-based agents with a market observer to adapt to various market conditions and minimize potential risks. Sun et al. [37] proposed a dynamic graph-based deep RL method that uses dynamic multi-channel graph attention to capture long- and short-term relationships based on static and dynamic graphs. Sun et al. [38] integrated GraphSAGE [44] and proximal policy optimization to capture the complex relationships between securities, market indices, and industry indices. Cheng and Sun [39] proposed a multi-agent portfolio adaptive trading framework that combines trading action and portfolio modules for long- and short-term situation judgment. However, these studies did not consider the asset allocation and security selection problems. In this paper, we not only select securities within each asset class but also determine investment ratios among asset classes.

C. ASSET ALLOCATION METHODS

Several asset allocation methods [8], [9], [10], [11], [12], [19] based on static and dynamic strategies have been proposed in the financial field. Faber [8] developed multiple global tactical asset allocation (GTAA) strategies, including GTAA5 and GTAA-Aggressive 3, based on absolute and relative momentum. Keller and Keuning [9] proposed a protective asset allocation (PAA) strategy based on relative and breadth momentum. These authors [10] also proposed multiple vigilant asset allocation (VAA) strategies, including VAA-G4 and VAA-G12, based on relative and breadth momentum. Subsequently, they [11] proposed a defensive asset allocation (DAA) strategy based on canary universe and breadth

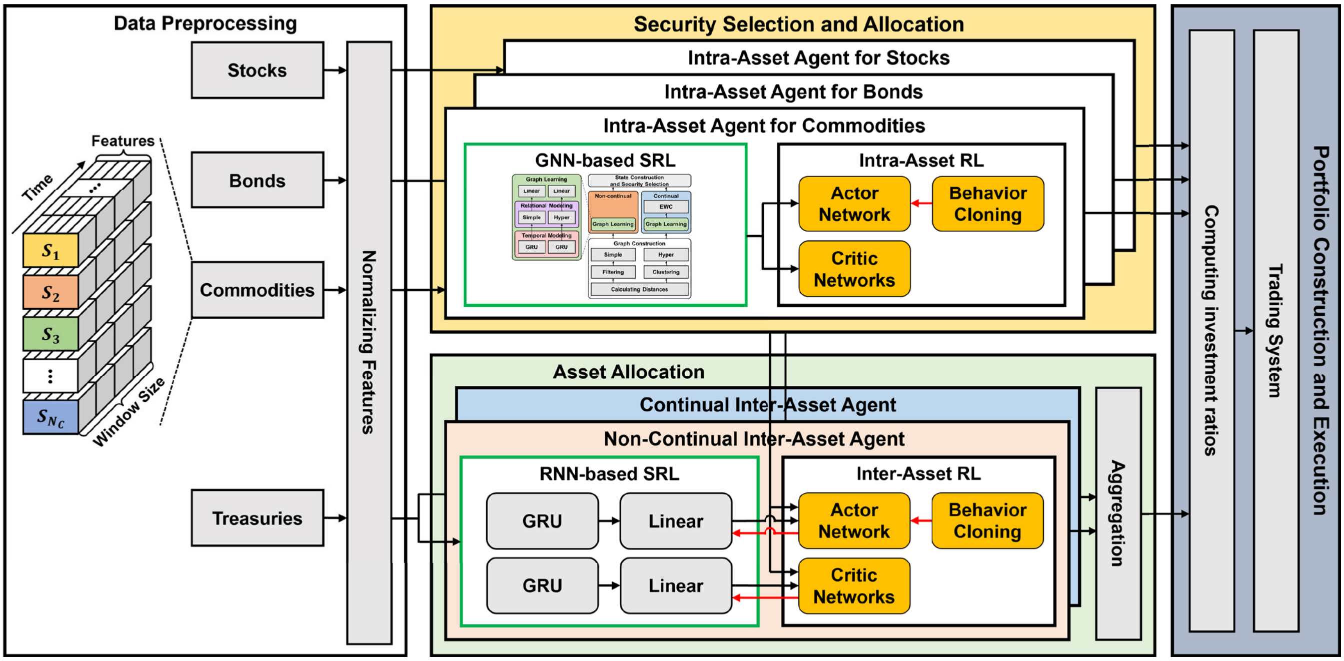


FIGURE 4. Architecture of MAMA.

momentum. Dalio [12] proposed the All-Weather strategy that allocates stocks, intermediate-term bonds, long-term bonds, gold, and commodities following fixed proportions. Ha and Fabozzi [19] proposed the dual momentum (DM) and composite DM (CDM) strategies based on absolute and relative momentum. However, these studies did not consider the security selection and security allocation problems. In this paper, we select securities within each asset class and determine their investment ratios.

III. PROPOSED MULTI-ASSET MULTI-AGENT RL FRAMEWORK

We propose a multi-asset multi-agent framework named MAMA that not only integrates asset allocation, security selection, and security allocation, but also considers concept drift and interest rates, thereby achieving fully autonomous and expert-level portfolio management.

A. ARCHITECTURE

Fig. 4 depicts the architecture of MAMA. The proposed architecture preprocesses security data corresponding to each asset class. For security selection and allocation, we apply a multi-agent structure that captures the risk-return characteristics of each asset class. Each intra-asset agent consists of GNN-based SRL and intra-asset RL. The GNN-based SRL selects profitable securities by accounting for concept drift at the security level. The continual graph learning in GNN-based SRL enables adaptation to changing graph structures (i.e., node and edge sets) while retaining knowledge. The intra-asset RL algorithm determines the investment ratios among the selected securities. For asset allocation, we integrate non-continual and continual inter-asset agents to handle concept drift at the asset class level. Each inter-asset agent consists of RNN-based SRL and inter-asset RL. The RNN-based SRL captures changes in interest rates by extracting temporal information from treasuries with various maturities, ranging from months to over 20 years. The inter-asset RL algorithm determines the investment ratios among asset classes. We leverage the TD3 algorithm as the intra-asset and inter-asset RL algorithms because it mitigates overestimation bias via clipped double Q-learning and enhances stability through target policy smoothing. TD3 algorithm often outperforms other RL algorithms in financial applications such as portfolio selection, optimization, and hedging [45], [46], [47]. Finally, the portfolio is constructed by computing the investment ratios of each selected security and cash, and then executed via a trading system. Each component is explained in detail in the following subsections.

B. DATA PREPROCESSING

We use only information on securities, including the opening, highest, lowest, closing, and volume values, for stocks, bonds, and commodities. Among bond securities, we use six securities (BIL, SGOV, IEF, VGIT, SHY, and TLT) as treasuries, covering a wide range of maturities from months to over 20 years. The input features of the securities are normalized using a robust standardization method [48], which reduces the influence of outliers using the interquartile range and median value. The normalized feature vectors for the stocks, bonds, and commodities are used as inputs for the GNN-based SRL, whereas those for the treasuries are used as inputs for the RNN-based SRL.

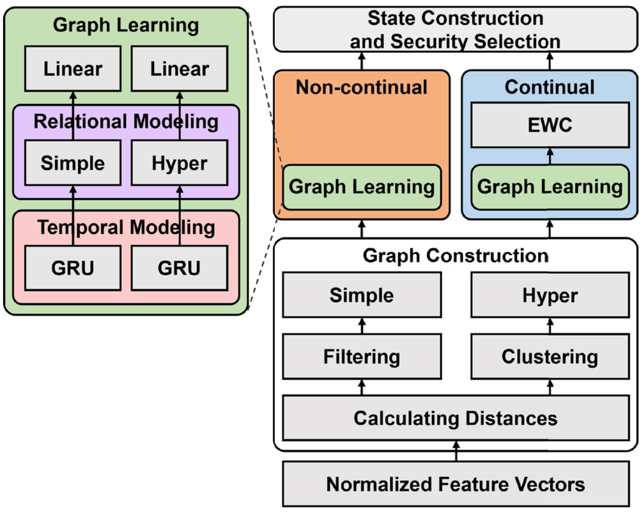


FIGURE 5. Architecture of GNN-based SRL.

C. SECURITY SELECTION AND ALLOCATION

For security selection and allocation, we propose a multi-agent structure in which each intra-asset agent is composed of GNN-based SRL and intra-asset RL, designed to select securities and determine their investment ratios.

1) GNN-BASED SRL

To address partial observability, we employ the GNN-based SRL to map historical observations (i.e., a sliding window) for the asset class to states. Fig. 5 shows the architecture of the GNN-based SRL. Simple graphs and hypergraphs are constructed based on normalized feature vectors. Subsequently, both the non-continual and continual models employ a graph learning method to learn both temporal and relational dependencies. Finally, states are constructed, and securities are selected simultaneously by integrating the non-continual and continual graph learning models. The constructed states are used as the input state for the intra- and inter-asset RL.

a: GRAPH CONSTRUCTION

For graph construction, we calculate the pairwise distances between securities using multi-dimensional dynamic time warping [49], and then construct simple graphs and hypergraphs based on these distances. We filter out insignificant edges in each simple graph to reduce noise. To this end, we sort the edges based on the associated distance and attach only the lower $\xi\%$ edges, where ξ is a hyperparameter. To ensure connectivity of the simple graph, we consider different values of ξ depending on the number of securities within an asset class. We set ξ to 20 if an asset class contains more than 50 securities, and to 40 otherwise. For each hypergraph, we generate hyperedges by clustering the target securities with the nearest securities based on distance. We leverage the K-nearest neighbor algorithm [50] to cluster securities, where K denotes a hyperparameter that controls the number of securities included in the hyperedge.

Algorithm 1 State Construction and Security Selection Algorithm

```
Input: (1) the non-continual ranking models $SG^j_{NC}$ and $HG^j_{NC}$, (2) the continual ranking models $SG^j_C$ and $HG^j_C$
Output: the average ranking score $R^j_A$ and selected security set $S^j_{selected}$
Aggregate the ranking scores of the simple graph-based ranking models:
    $R^j_{SG} \leftarrow AGGREGATE(SG^j_{NC}, SG^j_C)$
Aggregate the ranking scores of the hypergraph-based ranking models:
    $R^j_{HG} \leftarrow AGGREGATE(HG^j_{NC}, HG^j_C)$
Average the aggregated ranking scores for the simple graph and hypergraph: $R^j_A \leftarrow (R^j_{SG} + R^j_{HG}) / 2$
$S^j_{selected} \leftarrow topK_{SG}(R^j_{SG}) \cap topK_{HG}(R^j_{HG})$
if $|S^j_{selected}|$ is less than $K_A$ then
    $S^j_{selected} \leftarrow topK_A(R^j_A)$
end if
return $R^j_A, S^j_{selected}$
```

b: GRAPH LEARNING

The graph learning method comprises temporal and relational modeling. *a) Temporal modeling.* The input for the temporal modeling is a sliding window comprising feature vectors that are normalized via data preprocessing. We use one gated recurrent unit (GRU) layer to capture the temporal dependencies of individual securities, as shown in Fig. 5. *b) Relational modeling.* We apply GAT [51] for simple graphs and hypergraph convolution (HConv) with a multi-head attention mechanism [52] for hypergraphs to capture the relational features. To avoid oversmoothing, different numbers of GAT and HConv layers are used depending on the number of securities within an asset class. We use two layers each for GAT and HConv if an asset class contains more than 50 securities, and one layer each otherwise. Finally, a linear layer with a LeakyReLU activation function is used to predict the ranking score, which represents a ranked list of securities according to their expected return rates.

c: STATE CONSTRUCTION AND SECURITY SELECTION ALGORITHM

Algorithm 1 presents the state construction and security selection algorithm for an asset class j. For the simple graph and hypergraph, we combine non-continual and continual ranking models to ensure diversified adaptation at the security level. To achieve this, we integrate the ranking scores obtained from the ranking models (SG^j_{NC} and SG^j_C) for the simple graph and ranking models (HG^j_{NC} and HG^j_C)

for the hypergraph using an *AGGREGATE* function (lines 1 and 2). We leverage averaging as the *AGGREGATE* function. Subsequently, for the simple graph and hypergraph, we average the aggregated ranking scores (R^j_{SG} and R^j_{HG}) to construct the state R^j_A (line 3). Finally, we select the top-K_{SG} and top-K_{HG} securities based on the aggregated ranking scores, R^j_{SG} and R^j_{HG}, respectively, and compute their intersection to select the securities with the highest expected profits (line 4). If the number of securities included in the selected security set $S^j_{selected}$ is less than K_A, we select the top-K_A securities based on the average ranking score, R^j_A (line 6). The average ranking score vector R^j_A is used as the input state for the actor and critic networks in Algorithms 2 and 3.

We pretrain the GNN-based SRL to stabilize the learning process of intra-asset and inter-asset RL. To optimize non-continual ranking models SG^j_{NC} and HG^j_{NC}, as in [32], we introduce a loss function that integrates ranking loss and graph proximity loss. To optimize continual ranking models SG^j_C and HG^j_C, as in [32], we introduce an elastic weight consolidation (EWC) loss function [53], which limits parameter updates according to their importance.

2) INTRA-ASSET MARKOV DECISION PROCESS (MDP) FORMULATION

a: STATE

We define the intra-asset state s^j_t as the average ranking score vector R^j_A obtained from Algorithm 1.

b: ACTION

On each trading day t, the intra-asset agent for asset class j determines the intra-asset action a^j_t (i.e., the investment ratio among securities), subject to the constraint expressed by Equation (1), where S^j_{all} denotes the set of all securities contained in asset class j, S^j_t denotes the security set selected via Algorithm 1, and $a^{j,s}_t$ denotes the investment ratio of security s.

$$a^j_t = \{a^{j,s}_t\}_{s\in S^j_{all}}, \quad \text{where} \sum_{s\in S^j_t} a^{j,s}_t = 1, \quad a^{j,s}_t \geq 0 \tag{1}$$

c: REWARD

To maximize profits within an asset class j, we define the intra-asset reward r^j_t for the intra-asset action a^j_t as the logarithmic rate of return as in Equation (2), where RR^j_t denotes the return rate for asset class j on day t. The return rate RR^j_t for asset class j is calculated as the weighted sum of the change rate of the security closing price, as indicated by Equation (3), where $P^{j,s}_t$ denotes the closing price of security s on day t, and ξ denotes a transaction cost rate used to simulate a real trading environment.

$$r^j_t = \ln\left(\frac{RR^j_{t+1}}{RR^j_t}\right) \tag{2}$$

Algorithm 2 Intra-Asset RL Algorithm for Security Allocation

```
Input: Sliding-window data $w^j_t \leftarrow \{x^j_{t-N_w+1}, \ldots, x^j_{t-1}, x^j_t\}$ obtained from the data preprocessing
Initialize the pretrained GNN-based SRL network $\psi_{q^j}$ for asset class $j$
Initialize intra-asset critic networks $Q_{\theta^j_1}$, $Q_{\theta^j_2}$ for asset class $j$
Initialize an intra-asset actor network $\mu_{\phi^j}$ for asset class $j$

Initialize target networks: $\theta^{j'}_{1,2} \leftarrow \theta^j_{1,2}$, $\phi^{j'} \leftarrow \phi^j$
Initialize a replay buffer $\mathcal{D}$

for $e = 1$ to $N_{epochs}$ do
    Calculate the delay value for each epoch: $d \leftarrow (e \bmod \alpha) + \beta$

    for $t = 1$ to $T - 1$ do
        Select an intra-asset action $a^j_t$ with exploration noise $\epsilon^j \sim \mathcal{N}(0, \sigma)$:
            $a^j_t \leftarrow \mu_{\phi^j}(s^j_t) + \epsilon^j$ where $s^j_t = \psi_{q^j}(w^j_t)$
        Observe the intra-asset reward $r^j_t$ and next input $w^j_{t+1}$
        Store a transition $\langle w^j_t, a^j_t, r^j_t, w^j_{t+1} \rangle$ to $\mathcal{D}$

        Sample a mini-batch of $D$ transitions $\langle w^j_i, a^j_i, r^j_i, w^j_{i+1} \rangle$ from $\mathcal{D}$
        Smooth the intra-asset target policies with $\epsilon^j \sim \text{clip}(\mathcal{N}(0, \sigma'), -c, c)$:
            $\tilde{a}^j_{i+1} \leftarrow \mu_{\phi^{j'}}(s^j_{i+1}) + \epsilon^j$ where $s^j_{i+1} = \psi_{q^j}(w^j_{i+1})$
        $Y^j_{critic} \leftarrow r^j_i + \gamma \min_{l=1,2} Q_{\theta^{j'}_l}(s^j_{i+1}, \tilde{a}^j_{i+1})$,
            where $s^j_{i+1} = \psi_{q^j}(w^j_{i+1})$
        Update the intra-asset critics $\theta^j_l$ using the mean squared error (MSE) loss:
            $\frac{1}{D}\sum(Y^j_{critic} - Q_{\theta^j_l}(s^j_i, a^j_i))^2$ where $s^j_i = \psi_{q^j}(w^j_i)$

        if $t \bmod d$ then
            Update the intra-asset actor $\mu_{\phi^j}$ using the deterministic policy gradient:
                $\frac{1}{D}\sum \nabla Q_{\theta^j_1}(s^j_i, a^j_i)$ where $s^j_i = \psi_{q^j}(w^j_i)$,
                $a^j_i = \mu_{\phi^j}(\psi_{q^j}(w^j_t))$
            Update the intra-asset actor $\mu_{\phi^j}$ using the CE loss:
                $\frac{1}{D}\sum \nabla CE(a^j_i, a^j_{expert})$ where $a^j_i = \mu_{\phi^j}(\psi_{q^j}(w^j_t))$
            Soft-update the target networks:
                $\theta^{j'}_{1,2} \leftarrow \tau\theta^j_{1,2} + (1-\tau)\theta^{j'}_{1,2}$, $\phi^{j'} \leftarrow \tau\phi^j + (1-\tau)\phi^{j'}$
        end if
    end for
end for
```

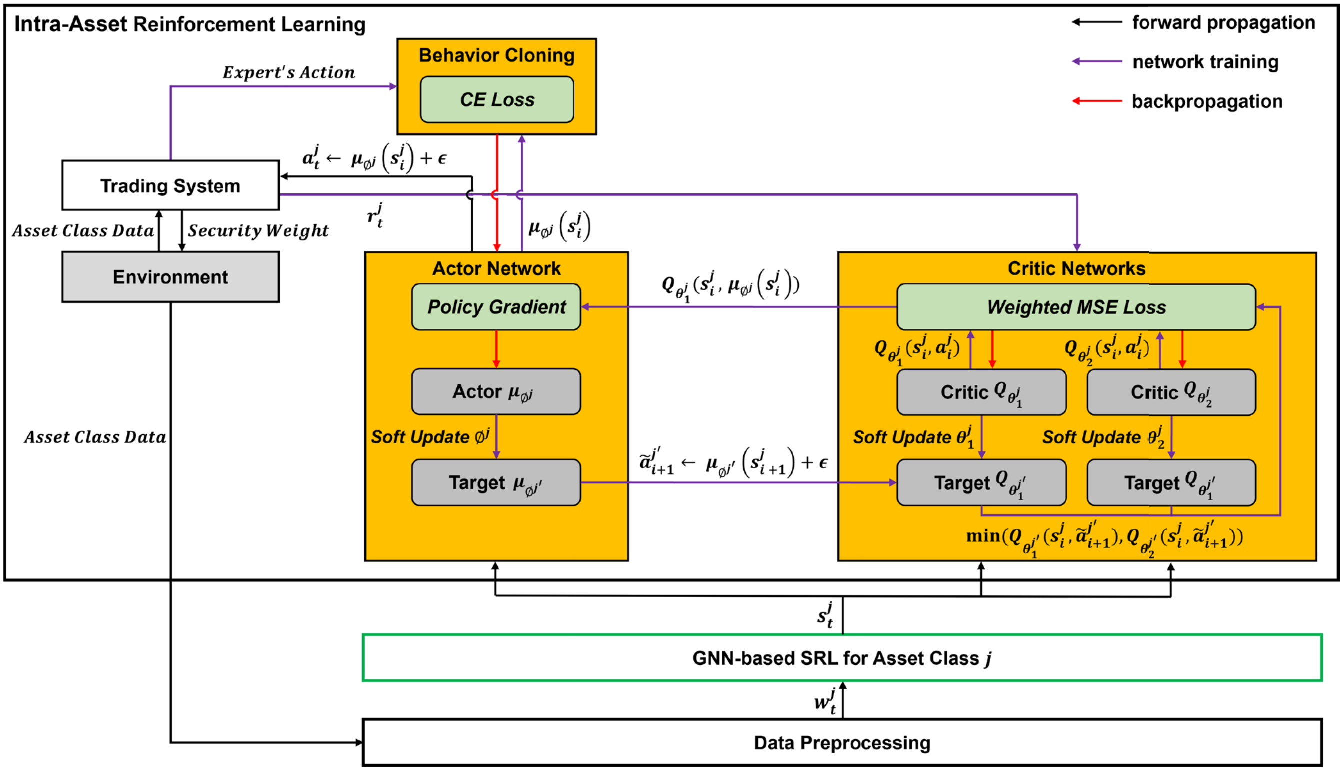

Intra-Asset Reinforcement Learning
forward propagation
network training
backpropagation
Behavior Cloning
CE Loss
Expert's Action
Trading System
$a_t^j \leftarrow \mu_{\emptyset^j}(s_i^j) + \epsilon$
r_t^j
$\mu_{\emptyset^j}(s_i^j)$
Asset Class Data
Security Weight
Environment
Actor Network
Policy Gradient
Actor $\mu_{\emptyset^j}$
Soft Update $\emptyset^j$
Target $\mu_{\emptyset^{j'}}$
$Q_{\theta_1^j}(s_i^j, \mu_{\emptyset^j}(s_i^j))$
$\tilde{a}_{i+1}^{j'} \leftarrow \mu_{\emptyset^{j'}}(s_{i+1}^j) + \epsilon$
Critic Networks
Weighted MSE Loss
$Q_{\theta_1^j}(s_i^j, a_i^j)$
$Q_{\theta_2^j}(s_i^j, a_i^j)$
Critic $Q_{\theta_1^j}$
Critic $Q_{\theta_2^j}$
Soft Update θ_1^j
Soft Update θ_2^j
Target $Q_{\theta_1^{j'}}$
Target $Q_{\theta_1^{j'}}$
$\min(Q_{\theta_1^{j'}}(s_i^j, \tilde{a}_{i+1}^{j'}), Q_{\theta_2^{j'}}(s_i^j, \tilde{a}_{i+1}^{j'}))$
Asset Class Data
s_t^j
GNN-based SRL for Asset Class j
w_t^j
Data Preprocessing

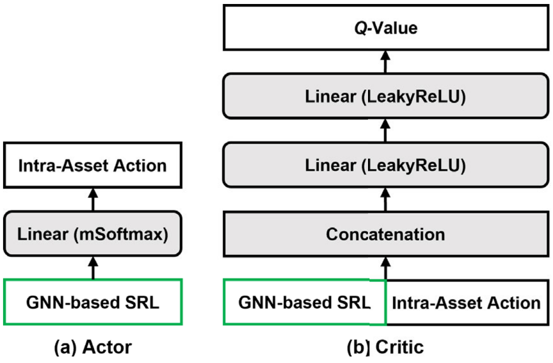
FIGURE 6. Architecture of the intra-asset RL.

$$RR_t^j = \sum_{s \in S_t^j} a_t^{j,s} \times \frac{P_{t+1}^{j,s} \times (1 - \xi) - P_t^{j,s}}{P_t^{j,s}} \tag{3}$$

3) INTRA-ASSET RL ALGORITHM

Algorithm 2 presents the intra-asset RL algorithm for asset class j. Fig. 6 depicts the architecture of the intra-asset RL. The intra-asset RL algorithm uses the pretrained GNN-based SRL network ψ_{q^j}, an actor network μ_{ϕ^j}, and two critic networks $Q_{\theta_1^j}$ and $Q_{\theta_2^j}$. All networks, except for the pretrained GNN-based SRL network, have corresponding target networks. After interacting with the environment, we store a transition of $\langle w_t^j, a_t^j, r_t^j, w_{t+1}^j \rangle$ to $\mathcal{D}$ (line 11).

In the intra-asset RL algorithm, the actor network μ_{ϕ^j} is combined with the pretrained GNN-based SRL network ψ_{q^j}. In the actor network μ_{ϕ^j} depicted in Fig. 7(a), we employ a masked softmax function (mSoftmax) at the output layer to determine the investment ratio only for the securities that are selected via Algorithm 1. We add a varying amount of noise to each output of the mSoftmax function, excluding zero outputs, for exploration (line 9). To avoid overfitting, we introduce the target policy smoothing of the original TD3 algorithm (line 13). Fig. 7(b) depicts the structures of the critic networks, $Q_{\theta_1^j}$ and $Q_{\theta_2^j}$. To mitigate overestimation, the clipped double Q-learning technique used in the original TD3 algorithm is adopted for calculating the target value Y_{critic}^j (line 14).



(a) Actor

(b) Critic

FIGURE 7. Neural network structures for intra-asset RL.

We adopt a behavior cloning technique to guide the training process of the actor network. On day t, the intra-asset expert selects the security with the highest one-day forward return rate from those selected via security selection. The one-day forward return rate $FR_t^{j,s}$ of each security is defined by Equation (4), where $P_t^{j,s}$ and $P_{t+1}^{j,s}$ denote the closing prices of the security s included in asset class j on the current and subsequent days, respectively. To learn the intra-asset action a_{expert}^j of the expert, the actor network is trained by minimizing the cross-entropy (CE) loss between the mSoftmax output vector a_i^j and the expert's intra-asset action a_{expert}^j encoded as a one-hot vector

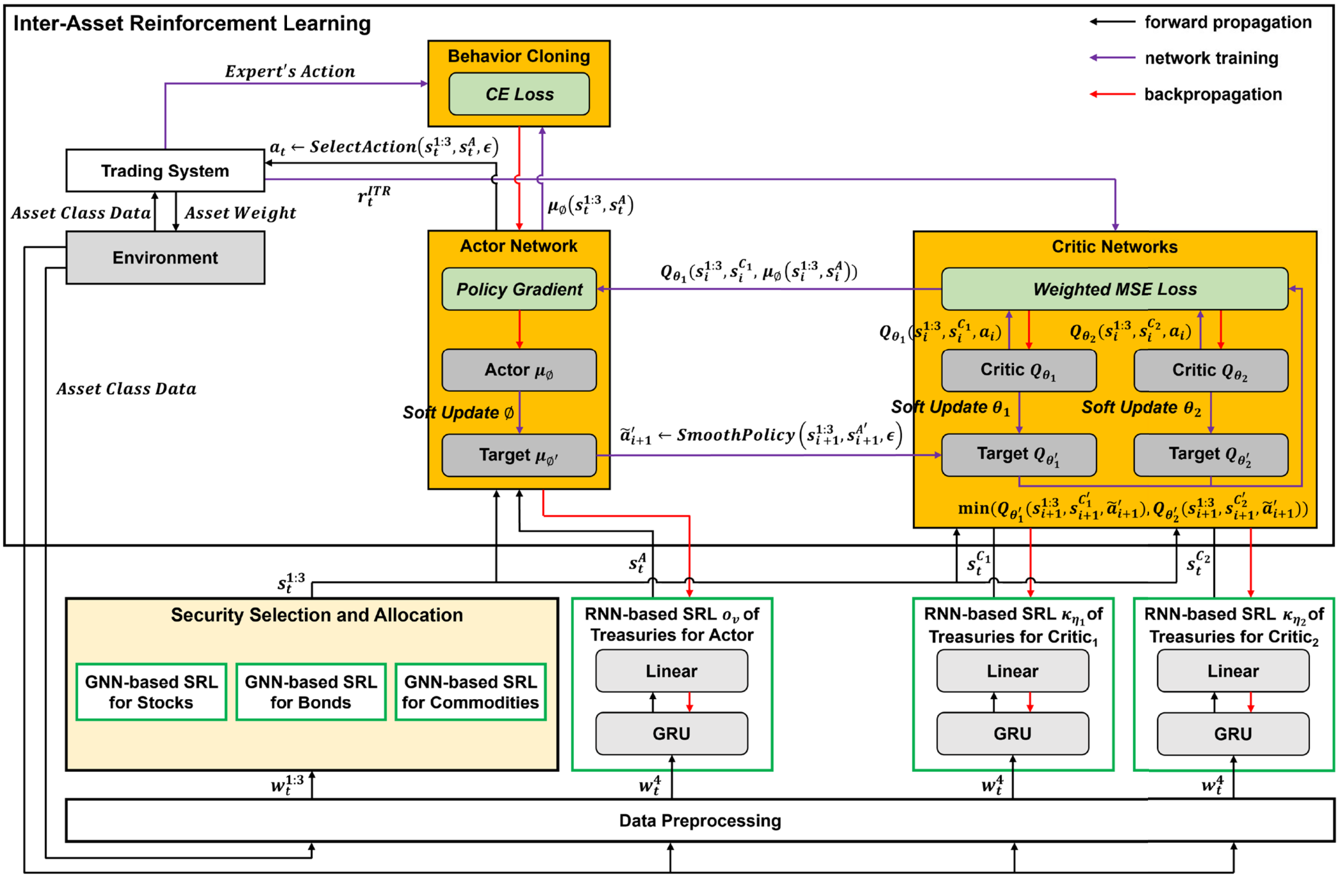
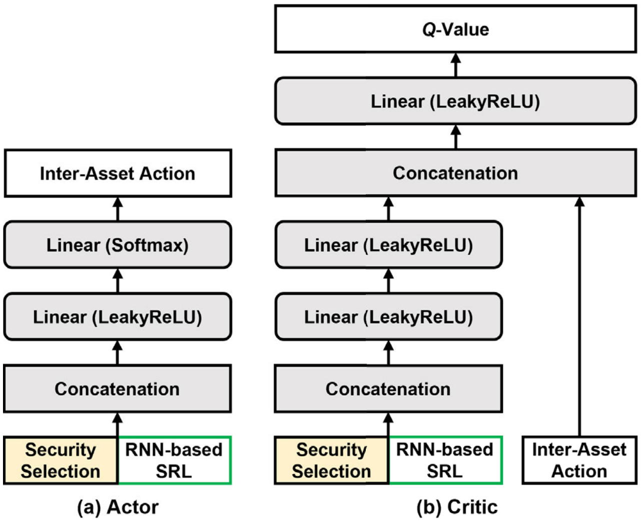

Inter-Asset Reinforcement Learning
forward propagation
network training
backpropagation
Behavior Cloning
CE Loss
Expert's Action
$a_t \leftarrow SelectAction(s_t^{1:3}, s_t^A, \epsilon)$
Trading System
r_t^{ITR}
Asset Class Data
Asset Weight
Environment
$\mu_\phi(s_t^{1:3}, s_t^A)$
Actor Network
Policy Gradient
Actor μ_ϕ
Soft Update ϕ
Target $\mu_{\phi'}$
$Q_{\theta_1}(s_i^{1:3}, s_i^{C_1}, \mu_\phi(s_i^{1:3}, s_i^A))$
Critic Networks
Weighted MSE Loss
$Q_{\theta_1}(s_i^{1:3}, s_i^{C_1}, a_i)$
$Q_{\theta_2}(s_i^{1:3}, s_i^{C_2}, a_i)$
Critic Q_{θ_1}
Critic Q_{θ_2}
Soft Update θ_1
Soft Update θ_2
Target $Q_{\theta_1'}$
Target $Q_{\theta_2'}$
$\tilde{a}_{i+1}' \leftarrow SmoothPolicy(s_{i+1}^{1:3}, s_{i+1}^{A'}, \epsilon)$
$\min(Q_{\theta_1'}(s_{i+1}^{1:3}, s_{i+1}^{C_1'}, \tilde{a}_{i+1}'), Q_{\theta_2'}(s_{i+1}^{1:3}, s_{i+1}^{C_2'}, \tilde{a}_{i+1}'))$
Asset Class Data
$s_t^{1:3}$
s_t^A
s_t^{C1}
s_t^{C2}
Security Selection and Allocation
GNN-based SRL for Stocks
GNN-based SRL for Bonds
GNN-based SRL for Commodities
RNN-based SRL o_v of Treasuries for Actor
RNN-based SRL κ_{η_1} of Treasuries for Critic$_1$
RNN-based SRL κ_{η_2} of Treasuries for Critic$_2$
Linear
GRU
$w_t^{1:3}$
w_t^4
Data Preprocessing

FIGURE 8. Architecture of inter-asset RL.



(a) Actor (b) Critic

FIGURE 9. Neural network structures for inter-asset RL.

(line 18).

$$FR_t^{j,s} = \frac{(P_{t+1}^{j,s} - P_t^{j,s})}{P_t^{j,s}} \tag{4}$$

The dynamic delay technique [3] is adopted to update the actor and target networks using varying delay values to enhance the training stability and efficiency. At each epoch, we calculate the delay value d using Equation (5), where α and β denote hyperparameters that control the variance and minimum delay value, respectively (line 7).

$$d = (e \bmod \alpha) + \beta \tag{5}$$

D. ASSET ALLOCATION

For asset allocation, we combine the non-continual and continual inter-asset agents, which are composed of RNN-based SRL and inter-asset RL, to determine the investment ratios among asset classes (i.e., stocks, bonds, commodities, and cash) while adapting to concept drift.

1) RNN-BASED SRL

To handle partial observability, we employ an RNN-based SRL to capture changes in interest rates by mapping historical observations (i.e., a sliding window) for treasuries to states. The input for the RNN-based SRL is a sliding window w_t^4 of the normalized data for securities included in the treasuries. As highlighted in Fig. 4, we apply one GRU layer to learn the temporal dependencies of individual securities and subsequently apply a linear layer that uses LeakyReLU as

Algorithm 3 Non-Continual and Continual Inter-Asset RL Algorithms for Asset Allocation

```
Input: Sliding-window data $w_t^{1:4} \leftarrow \{x_{t-N_w+1}^j, \dots, x_{t-1}^j, x_t^j\}_{j=1}^4$ obtained from the data preprocessing for all asset classes
Initialize the pretrained GNN-based SRL networks $\psi_{q^{1:3}} \leftarrow \{\psi_{q^j}\}_{j=1}^3$ for stocks, bonds, and commodities
Initialize the non-continual inter-asset critic networks $Q_{\theta_1^{NC}}, Q_{\theta_2^{NC}}$, RNN-based SRL network $\kappa_{\eta_1^{NC}}, \kappa_{\eta_2^{NC}}$ for $Q_{\theta_1^{NC}}, Q_{\theta_2^{NC}}$
Initialize the non-continual inter-asset actor network $\mu_{\phi^{NC}}$, RNN-based SRL network $o_{\nu^{NC}}$ for $\mu_{\phi^{NC}}$
Initialize the continual inter-asset critic networks $Q_{\theta_1^C}, Q_{\theta_2^C}$, RNN-based SRL networks $\kappa_{\eta_1^C}, \kappa_{\eta_2^C}$ for $Q_{\theta_1^C}, Q_{\theta_2^C}$
Initialize the continual inter-asset actor network $\mu_{\phi^C}$, RNN-based SRL network $o_{\nu^C}$ for $\mu_{\phi^C}$
Initialize target networks: $\theta_{1,2}^{NC'} \leftarrow \theta_{1,2}^{NC}, \eta_{1,2}^{NC'} \leftarrow \eta_{1,2}^{NC}, \phi^{NC'} \leftarrow \phi^{NC}, \nu^{NC'} \leftarrow \nu^{NC}, \theta_{1,2}^{C'} \leftarrow \theta_{1,2}^{C}, \eta_{1,2}^{C'} \leftarrow \eta_{1,2}^{C}, \phi^{C'} \leftarrow \phi^{C}, \nu^{C'} \leftarrow \nu^{C}$
Initialize a replay buffer $\mathcal{D}$

for $e = 1$ to $N_{epochs}$ do
    Calculate the delay value for each epoch: $d \leftarrow (e \bmod \alpha) + \beta$

    for $t = 1$ to $T - 1$ do
        Select a non-continual inter-asset action $a_t^{NC}$ with exploration noise $\epsilon^{NC} \sim \mathcal{N}(0, \sigma)$:
            $a_t^{NC} \leftarrow SelectAction(s_t^{1:3}, s_t^{NC,A}, \epsilon^{NC})$ where $s_t^{1:3} = \psi_{q^{1:3}}(w_t^{1:3}), s_t^{NC,A} = o_{\nu^{NC}}(w_t^4)$
        Select a continual inter-asset action $a_t^C$ with exploration noise $\epsilon^C \sim \mathcal{N}(0, \sigma)$:
            $a_t^C \leftarrow SelectAction(s_t^{1:3}, s_t^{C,A}, \epsilon^C)$ where $s_t^{1:3} = \psi_{q^{1:3}}(w_t^{1:3}), s_t^{C,A} = o_{\nu^C}(w_t^4)$

        Observe the inter-asset reward $r_t^{ITR}$ and the next input $w_{t+1}^{1:4}$
        Store a transition $\langle w_t^{1:4}, a_t^{NC}, a_t^C, r_t^{ITR}, w_{t+1}^{1:4} \rangle$ to $\mathcal{D}$

        Sample a mini-batch of $D$ transitions $\langle w_i^{1:4}, a_i^{NC}, a_i^C, r_i^{ITR}, w_{i+1}^{1:4} \rangle$ from $\mathcal{D}$
        Smooth the non-continual inter-asset target policy with $\epsilon^{NC} \sim \text{clip}(\mathcal{N}(0, \sigma'), -c, c)$:
            $\tilde{a}_{i+1}^{NC'} \leftarrow SmoothPolicy(s_{i+1}^{1:3}, s_{i+1}^{NC,A'}, \epsilon^{NC})$ where $s_t^{1:3} = \psi_{q^{1:3}}(w_t^{1:3}), s_{i+1}^{NC,A'} = o_{\nu^{NC'}}(w_{i+1}^4)$
        $Y_{critic}^{NC} \leftarrow CalculateTarget(r_i^{ITR}, s_{i+1}^{1:3}, s_{i+1}^{NC,C_l'}, \tilde{a}_{i+1}^{NC'})$ where $s_t^{1:3} = \psi_{q^{1:3}}(w_t^{1:3}), s_{i+1}^{NC,C_l'} = \kappa_{\eta_l^{NC'}}(w_{i+1}^4)$
        Update the non-continual inter-asset critics $\theta_l^{NC}$ using the MSE loss:
            $UpdateCritics(Y_{critic}^{NC}, s_i^{1:3}, s_i^{NC,C_l}, a_i^{NC})$ where $s_t^{1:3} = \psi_{q^{1:3}}(w_t^{1:3}), s_i^{NC,C_l} = \kappa_{\eta_l^{NC}}(w_i^4)$

        Smooth the continual inter-asset target policy with $\epsilon^C \sim \text{clip}(\mathcal{N}(0, \sigma'), -c, c)$:
            $\tilde{a}_{i+1}^{C'} \leftarrow SmoothPolicy(s_{i+1}^{1:3}, s_{i+1}^{C,A'}, \epsilon^C)$ where $s_t^{1:3} = \psi_{q^{1:3}}(w_t^{1:3}), s_{i+1}^{C,A'} = o_{\nu^{C'}}(w_{i+1}^4)$
        $Y_{critic}^{C} \leftarrow CalculateTarget(r_i^{ITR}, s_{i+1}^{1:3}, s_{i+1}^{C,C_l'}, \tilde{a}_{i+1}^{C'})$ where $s_t^{1:3} = \psi_{q^{1:3}}(w_t^{1:3}), s_{i+1}^{C,C_l'} = \kappa_{\eta_l^{C'}}(w_{i+1}^4)$
        Update the continual inter-asset critics $\theta_l^C$ using the EWC loss [53]:
            $UpdateCritics(Y_{critic}^{C}, s_i^{1:3}, s_i^{C,C_l}, a_i^C) + \frac{\lambda}{2} \sum_h F_h(\theta_{l,h}^C - \theta_{old,l,h}^{C*})$ where $s_t^{1:3} = \psi_{q^{1:3}}(w_t^{1:3}), s_i^{C,C_l} = \kappa_{\eta_l^C}(w_i^4)$

        if $t \bmod d$ then
            Update the non-continual inter-asset actor $\mu_{\phi^{NC}}$ using the deterministic policy gradient:
                $UpdateActor(s_i^{1:3}, s_i^{NC,C_1}, a_i^{NC})$ where $s_t^{1:3} = \psi_{q^{1:3}}(w_t^{1:3}), s_i^{NC,C_1} = \kappa_{\eta_1^{NC}}(w_i^4), a_i^{NC} = \mu_{\phi^{NC}}(s_i^{1:3}, o_{\nu^{NC}}(w_i^4))$
            Update the non-continual inter-asset actor $\mu_{\phi^{NC}}$ using the CE loss for the behavior cloning:
                $UpdateActorWithBC(a_i^{NC}, a_{expert}^{ITR})$ where $a_i^{NC} = \mu_{\phi^{NC}}(\psi_{q^{1:3}}(w_t^{1:3}), o_{\nu^{NC}}(w_i^4))$
            Update the continual inter-asset actor $\mu_{\phi^C}$ using the EWC loss [53]:
                $UpdateActor(s_i^{1:3}, s_i^{C,C_1}, a_i^C) + \frac{\lambda}{2} \sum_h F_h(\phi_h^C - \phi_{old,h}^{C*})$ where $s_t^{1:3} = \psi_{q^{1:3}}(w_t^{1:3}), s_i^{C,C_1} = \kappa_{\eta_1^C}(w_i^4)$,
                                                                        $a_i^C = \mu_{\phi^C}(s_i^{1:3}, o_{\nu^C}(w_i^4))$
            Update the continual inter-asset actor $\mu_{\phi^C}$ using the CE loss for the behavior cloning:
                $UpdateActorWithBC(a_i^C, a_{expert}^{ITR})$ where $a_i^C = \mu_{\phi^C}(\psi_{q^{1:3}}(w_t^{1:3}), o_{\nu^C}(w_i^4))$
            Soft-update the target networks: $\theta_{1,2}^{NC'} \leftarrow \tau\theta_{1,2}^{NC} + (1-\tau)\theta_{1,2}^{NC'}, \eta_{1,2}^{NC'} \leftarrow \tau\eta_{1,2}^{NC} + (1-\tau)\eta_{1,2}^{NC'}, \phi^{NC'} \leftarrow \tau\phi^{NC} + (1-\tau)\phi^{NC'}$,
                $\nu^{NC'} \leftarrow \tau\nu^{NC} + (1-\tau)\nu^{NC'}, \theta_{1,2}^{C'} \leftarrow \tau\theta_{1,2}^{C} + (1-\tau)\theta_{1,2}^{C'}, \eta_{1,2}^{C'} \leftarrow \tau\eta_{1,2}^{C} + (1-\tau)\eta_{1,2}^{C'}$,
                $\phi^{C'} \leftarrow \tau\phi^{C} + (1-\tau)\phi^{C'}, \nu^{C'} \leftarrow \tau\nu^{C} + (1-\tau)\nu^{C'}$
        end if
    end for
end for
```

the activation function. The output of the linear layer is used as the input state for the actor and critic networks in the inter-asset RL. Each actor and each critic network is combined with its corresponding RNN-based SRL network. The RNN-based SRL networks are jointly trained with the actor and critic networks.

2) INTER-ASSET MDP FORMULATION

a: STATE

For the non-continual and continual inter-asset agent, we define the inter-asset state by concatenating the outputs $s_t^{1:3}$ of the GNN-based SRL for stocks, bonds, and commodities with the output of the RNN-based SRL for treasuries.

b: ACTION

On each trading day t, the non-continual and continual inter-asset agents determine the non-continual and continual inter-asset actions a_t^{NC} and a_t^{C} (i.e., the investment ratio among asset classes), respectively, each of which is subject to its corresponding constraint shown in Equations (6) and (7), where $a_t^{NC,j}$ and $a_t^{C,j}$ denote the investment ratios for asset class j.

$$a_t^{NC} = \{a_t^{NC,j}\}_{j=1}^{4}, \quad \text{where} \sum_{j=1}^{4} a_t^{NC,j} = 1, \quad a_t^{NC,j} \geq 0 \tag{6}$$

$$a_t^{C} = \{a_t^{C,j}\}_{j=1}^{4}, \quad \text{where} \sum_{j=1}^{4} a_t^{C,j} = 1, \quad a_t^{C,j} \geq 0 \tag{7}$$

c: REWARD

To minimize downside risk of the portfolio while maximizing long-term growth, we define the inter-asset reward r_t^{ITR} for the non-continual and continual inter-asset actions (a_t^{NC} and a_t^{C}) as the Calmar ratio as in Equation (8). In Equation (8), the CAGR is defined using the total portfolio value PV_t on day t and initial portfolio value PV_0, as determined by Equation (9), where day t is annualized by division by 252, which indicates the average number of trading days in a year. The total portfolio value PV_t represents the sum of the portfolio values of all asset classes, as determined by Equation (10), where PV_t^{cash} denotes the portfolio value of cash, indicating the remaining cash balance. The portfolio value PV_t^j of all asset classes except for cash is calculated using Equation (11), where $n^{j,s}$ denotes the number of owned shares corresponding to the security s included in asset class j. The maximum drawdown (MDD) represents the maximum percentage decline from the peak to the trough of a portfolio over a specified period T, as given by Equation (12). The drawdown for time τ is calculated using the inner maximum term.

$$r_t^{ITR} = \frac{CAGR(t+1)}{MDD(t+1)} \tag{8}$$

$$CAGR(t) = (\frac{PV_t}{PV_0})^{\frac{1}{t/252}} - 1 \tag{9}$$

$$PV_t = \sum_{j=1}^{3} PV_t^j + PV_t^{cash} \tag{10}$$

$$PV_t^j = \sum_{s \in S_t^j} n^{j,s} \times P_t^{j,s} \times (1 - \xi) \tag{11}$$

$$\text{MDD}(T) = \max_{\tau \in (0,T)} \left[\max_{t \in (0,\tau)} \frac{PV_t - PV_\tau}{PV_t} \right] \tag{12}$$

Algorithm 4 Portfolio Construction Algorithm

```
Input: (1) non-continual and continual inter-asset actions $a_t^{NC}$ and $a_t^{C}$,
       (2) intra-asset actions $a_t^{1:3}$ and the selected security set $S_{selected}^{1:3}$
Output: the portfolio $PORT_t$

Aggregate the non-continual and continual inter-asset actions:
    $a_t^A \leftarrow AGGREGATE(a_t^{NC}, a_t^{C})$
for $j \in \{\text{stocks, bonds, commodities}\}$ do
    for $s \in S_{selected}^j$ do
        Compute the final investment ratio: $IR_t^s \leftarrow a_t^{j,s} \times a_t^{A,j}$
        Append the tuple of the security and its investment ratio, $(s, IR_t^s)$, to $PORT_t$
    end for
end for
return $PORT_t$
```

3) INTER-ASSET RL ALGORITHM

Algorithm 3 presents the non-continual and continual inter-asset RL algorithms. Fig. 8 depicts the architecture of the inter-asset RL. The non-continual inter-asset RL algorithm uses an actor network $\mu_{\phi^{NC}}$, an RNN-based SRL network $o_{\nu^{NC}}$ for $\mu_{\phi^{NC}}$, two critic networks $Q_{\theta_1^{NC}}$ and $Q_{\theta_2^{NC}}$, and two RNN-based SRL networks $\kappa_{\eta_1^{NC}}$ and $\kappa_{\eta_2^{NC}}$ for $Q_{\theta_1^{NC}}$ and $Q_{\theta_2^{NC}}$. The continual inter-asset RL algorithm uses an actor network $\mu_{\phi^{C}}$, an RNN-based SRL network $o_{\nu^{C}}$ for $\mu_{\phi^{C}}$, two critic networks $Q_{\theta_1^{C}}$ and $Q_{\theta_2^{C}}$, and two RNN-based SRL networks $\kappa_{\eta_1^{C}}$ and $\kappa_{\eta_2^{C}}$ for $Q_{\theta_1^{C}}$ and $Q_{\theta_2^{C}}$. The pretrained GNN-based SRL networks $\psi_{q^{1:3}} = \{\psi_{q^j}\}_{j=1}^{3}$ are shared by both the non-continual and continual inter-asset RL algorithms. All networks, except for the pretrained GNN-based SRL networks, have corresponding target networks. The inputs for the pretrained GNN-based SRL networks are the sliding window $w_t^{1:3}$ obtained from the preprocessed security data corresponding to the stocks, bonds, and commodities. After interacting with the environment, we store a transition of $\langle w_t^{1:4}, a_t^{NC}, a_t^{C}, r_t^{ITR}, w_{t+1}^{1:4} \rangle$ to $\mathcal{D}$.

In the non-continual inter-asset RL algorithm, the actor network $\mu_{\phi^{NC}}$, depicted in Fig. 9(a), is integrated with the pretrained GNN-based SRL networks $\psi_{q^{1:3}}$ and RNN-based SRL network $o_{\nu^{NC}}$. In the actor network depicted in Fig. 9(a), we employ a softmax function at the output layer to determine the investment ratio corresponding to the asset classes. We select a non-continual inter-asset action a_t^{NC} using the *SelectAction* function (defined in Equation (13)), which adds a varying amount of noise to each output of the softmax function for exploration (line 11). To avoid overfitting, we smooth the non-continual inter-asset target policy using the *SmoothPolicy* function defined by Equation (14) (line 16). The target policy smoothing technique used in the original TD3 algorithm is adopted as the *SmoothPolicy* function. Whenever the actor network is updated using the *UpdateActor* function defined by Equation (15), the corresponding RNN-based SRL network is also updated via backpropagation.

$$SelectAction(s_t^{1:3}, s_t^A, \epsilon) = \mu_\phi(s_t^{1:3}, s_t^A) + \epsilon \tag{13}$$

$$SmoothPolicy(s_{i+1}^{1:3}, s_{i+1}^{A'}, \epsilon) = \mu_{\phi'}(s_{i+1}^{1:3}, s_{i+1}^{A'}) + \epsilon \tag{14}$$

$$UpdateActor(s_i^{1:3}, s_i^{C_1}, a_i) = \frac{1}{D}\sum \nabla Q_{\theta_1}(s_i^{1:3}, s_i^{C_1}, a_i) \tag{15}$$

Each critic network $Q_{\theta_l^{NC}}$, depicted in Fig. 9(b), is integrated with the pretrained GNN-based SRL networks $\psi_{q^{1:3}}$ and RNN-based SRL network $\kappa_{\eta_l^{NC}}$. To mitigate overestimation, the clipped double Q-learning technique used in the original TD3 algorithm is adopted for calculating the target value Y_{critic}^{NC}. The target value Y_{critic}^{NC} is calculated using the *CalculateTarget* function defined by Equation (16) (line 20). Whenever each critic network is updated using the *UpdateCritics* function defined by Equation (17), the corresponding RNN-based SRL network is also updated via backpropagation.

$$\begin{aligned} &CalculateTarget(r_i^{ITR}, s_{i+1}^{1:3}, s_{i+1}^{C_l'}, \tilde{a}'_{i+1}) \\ &= r_i^{ITR} + \gamma \min_{l=1,2} Q_{\theta_l'}(s_{i+1}^{1:3}, s_{i+1}^{C_l'}, \tilde{a}'_{i+1}) \end{aligned} \tag{16}$$

$$\begin{aligned} &UpdateCritics(Y_{critic}, s_i^{1:3}, s_i^{C_l}, a_i) \\ &= \frac{1}{D}\sum (Y_{critic} - Q_{\theta_l}(s_i^{1:3}, s_i^{C_l}, a_i))^2 \end{aligned} \tag{17}$$

We adopt a behavior cloning technique similar to that in the intra-asset RL algorithm. On day t, the inter-asset expert calculates the average one-day forward return rate AFR_t^j of each asset class using $FR_t^{j,s}$ defined by Equation (4), as shown in Equation (18). Subsequently, the inter-asset expert selects the asset class with the highest average one-day forward return rate. To learn the inter-asset action a_{expert}^{ITR} of the expert, we train the actor network using the *UpdateActorWithBC* function, as defined by Equation (19) (line 24). The *UpdateActorWithBC* function minimizes the CE loss between the softmax output vector a_i^{NC} and the expert's inter-asset action a_{expert}^{ITR} encoded as a one-hot vector. Finally, we adopt the dynamic delay technique, as in Algorithm 2 (line 7).

$$AFR_t^j = \frac{1}{|S_{all}^j|}\sum_{s \in S_{all}^j} FR_t^{j,s} \tag{18}$$

$$\begin{aligned} &UpdateActorWithBC(a_i, a_{expert}^{ITR}) \\ &= \frac{1}{D}\sum \nabla CE(a_i, a_{expert}^{ITR}) \end{aligned} \tag{19}$$

The continual inter-asset RL algorithm (highlighted in red in Algorithm 3) operates identically to the non-continual inter-asset RL algorithm, except for the update processes of the actor and critic networks. We employ the EWC loss function [53] to update the actor and critic networks (lines 21 and 25).

TABLE 1. Comparison of state-of-the-art methods.

	Security Selection		Security Allocation		Asset Allocation		
	ranking	continual	behavior cloning	dynamic delay	behavior cloning	dynamic delay	continual
MAMA	○	○	○	○	○	○	○
DASS [32]	○	○	×	×	×	×	×
GRL [38]	×	×	×	×	×	×	×
MASA [36]	×	×	×	×	×	×	×

E. PORTFOLIO CONSTRUCTION AND EXECUTION

Algorithm 4 presents the portfolio construction algorithm. The intra-asset and inter-asset actions indicate the investment ratios across the selected securities and asset classes, respectively. We aggregate the non-continual and continual inter-asset actions a_t^{NC} and a_t^C using the *AGGREGATE* function (line 1). We leverage averaging as the *AGGREGATE* function. Subsequently, for stocks, bonds, and commodities, we compute the final investment ratio IR_t^s of each selected security as the product of its investment ratio $a_t^{j,s}$ and the aggregated investment ratio $a_t^{A,j}$ for asset class j to which it belongs (line 4). We append the tuple of the security s and its final investment ratio IR_t^s to the portfolio $PORT_t$ (line 5). Finally, we execute the portfolio $PORT_t$ through a trading system.

F. DISCUSSION

Table 1 compares the proposed MAMA framework with state-of-the-art DL- and RL-based methods in terms of security selection, security allocation, and asset allocation. In Table 1, the notation ○ indicates that the corresponding technique is applied, whereas × indicates that it is not. To the best of our knowledge, MAMA is the only method that integrates security selection, security allocation, and asset allocation to achieve fully autonomous and expert-level portfolio management. Moreover, MAMA is an all-encompassing solution that effectively combines various techniques (i.e., the ranking approach, behavior cloning, dynamic delay, and continual learning) for the security selection, security allocation, and asset allocation dimensions. Although MAMA adopts some techniques used in previous methods, combining these techniques to achieve positive results is not straightforward.

IV. EXPERIMENTS

A. EXPERIMENTAL SETUP

1) DATASETS

We evaluated the proposed MAMA framework using asset classes: stocks, bonds (including treasuries), and commodities. For each asset class, we collected the securities based on the start date of the test period for each phase. For stocks, we used securities included in the S&P 100 index. For bonds, we selected the top 100 securities in terms of market capitalization from those categorized as bonds in an ETF

TABLE 2. Summary of comparison methods.

Strategies		Methods
Passive Strategy		Buy and Hold (B&H)
Security Selection		DASS [32], DASS+L
Security Allocation		MASA-DC [36], MASA-LSTM [36], MASA-MLP [36], GRL [38]
Asset Allocation	Static Strategies	All-Weather
	Dynamic Strategies	GTAA5, GTAA-Aggressive 3 (GTAA-Agg3), PAA, VAA-G4, VAA-G12, DAA, DM, CDM
Portfolio Selection	Benchmark Strategies	Constant Rebalanced Portfolio (CRP) [54], Best CRP [54]
	Follow-the-Winner Strategies	Universal Portfolios [54], Exponential Gradient (EG) [55]
	Follow-the-Loser Strategies	Anticorr [56], Passive Aggressive Mean Reversion [57], Online Moving Average Reversion (OLMAR) [58], Confidence Weighted Mean Reversion [59]
	Pattern Matching Strategies	Nonparametric Kernel based Log Optimal Strategy [60], Correlation-driven Nonparametric Learning (CORN) [61]

database [62]. For commodities, we selected 34 securities with a market capitalization exceeding \$90 million from those categorized as commodities in the ETF database. For treasuries, we selected six securities (BIL, SGOV, IEF, VGIT, SHY, and TLT) among securities included in bonds, each with a different maturity range. We collected data corresponding to securities from Yahoo Finance [63], including the opening, highest, lowest, closing, and volume values. The proposed framework was tested under various market conditions using the walk-forward testing method [64], as illustrated in Fig. 2. This method partitions the test data into segments corresponding to distinct phases. On average, the training and testing periods per phase were 502 and 124 days, respectively. All experiments were conducted three times per phase, and the average results were reported.

To simulate real-world trading conditions, we started with an initial balance of \$10,000 and only considered long positions unless stated otherwise. We purchased the maximum available number of shares for each security and sold them at the closing price on the following day for the sake of simplicity.

2) EVALUATION METRICS

To evaluate the performance of each method, we used the return rate and risk indicators (i.e., the Sharpe ratio—SR [65] and MDD defined by Equation (12)). The return rate was calculated as the ratio of portfolio value PV_{end} after the test period and initial balance PV_{start} as $(PV_{end} - PV_{start})/PV_{start}$.

The SR measures the return of an investment with respect to its risk [3], as expressed by Equation (20), where $\mathbb{E}[R]$ indicates the expected return and $\sigma[R]$ indicates the standard deviation of the return, which represents the fluctuation (i.e., risk). In Equation (20), we adopt the portfolio change rate as the return.

$$\mathrm{SR} = \frac{\mathbb{E}[R]}{\sigma[R]} \tag{20}$$

3) COMPARISON METHODS

The performance of the proposed MAMA method was compared with those of state-of-the-art methods, as shown in Table 2. The portfolio selection methods in Table 2 determine the investment ratios among a given set of securities using algorithmic strategies. To ensure a fair comparison, we extended all methods, except for the asset allocation methods, to multi-asset classes (i.e., stocks, bonds, commodities, and cash) and allocated equal proportions to each asset class. For the asset allocation and portfolio selection methods, we reported only the result of the best-performing method for each strategy. We rebalanced the asset allocation methods on a monthly basis. The Adam optimizer [66] was used with momentum parameters $\beta_1 = 0.9$ and $\beta_2 = 0.999$, an epsilon of 10^{-7}, a decay of 0.99, and a mini-batch size of 16. The network architectures and hyperparameters for each method were optimized as described below. The other parameters were considered to be identical to those described in the original studies.

- **B&H** buys SPY, BND, DBC, and cash in equal proportions on the first day of each test phase, holds them, and sells them on the last day of the test phase.
- **DASS** leverages a graph learning method consisting of low-level temporal, relational, and high-level temporal modeling and incorporates non-continual and continual graph learning.
- **DASS+L** is an enhanced DASS method that implements simple security allocation, linearly increasing the investment ratio based on the ranking score of the securities that are selected via DASS.
- **MASA-DC** leverages directional changes as the market observer, TD3, and controller barrier function method [67]. The actor network comprises two LeakyReLU dense layers with 400 and 300 units, and a softmax output layer. The critic networks comprise two LeakyReLU dense layers with 400 and 300 units, and a tanh output layer. We set the noise size for the exploration and regularization to 0.2, the clipping size to 1, and the learning rate to 0.0005.
- **MASA-LSTM** is a variant of MASA-DC, which leverages the LSTM layer as the market observer.
- **MASA-MLP** is a variant of MASA-DC that leverages MLP layer as the market observer.
- **GRL** leverages GraphSAGE and proximal policy optimization. The actor network comprises two LeakyReLU dense layers with 128 units and a softmax output layer. The critic network comprises two LeakyReLU dense layers with 128 units and a LeakyReLU output layer.

TABLE 3. Hyperparameter values.

Section	Hyperparameter	Value
Data preprocessing	Sliding window size	8
Security selection (Algorithm 1)	Number of top securities to select (simple graph) (K_{SG})	5
	Number of top securities to select (hypergraph) (K_{HG})	5
	Minimum number of securities to select (K_A)	2
Security allocation and asset allocation (Algorithms 2 and 3)	Variance delay value (α)	4
	Minimum delay value (β)	2
	Noise for exploration (σ)	0.2
	Noise for regularization (σ')	0.2
	Noise clipping size (c)	1
	Regularization strength in EWC loss (λ)	0.5
	Batch size	16
	Learning rate	0.0005
Trading	Transaction cost rate in Equations (3) and (11) (ξ)	0.1%

We set the learning rate to 0.001 and the number of K epochs to 20.

- **All-Weather** invests in SPY, TLT, IEF, GLD, and DBC with fixed proportions of 30%, 40%, 15%, 7.5%, and 7.5%, respectively.
- **GTAA-Agg3** calculates the average value of 1, 3, 6, and 12 month return rates for each ETF and invests in the three ETFs with the highest average return rate in the same proportions. Among the three ETFs, those with closing price lower than their 10-month moving average are held as cash.
- **CRP** is a benchmark strategy that invests in all securities with equal proportions.
- **EG** is a follow-the-winner strategy that increases the investment ratios of securities with the most recent best performance among the securities.
- **OLMAR** is a follow-the-loser strategy that increases the investment ratios for securities with the worst recent performance among the securities.
- **CORN** is a pattern matching strategy that determines the investment ratios among securities based on the historical price patterns.

4) IMPLEMENTATION DETAILS OF MAMA

Table 3 summarizes the hyperparameter values used in this paper. Each value in Table 3 was chosen empirically by testing a range of values for each hyperparameter. We conducted experiments by varying the values of major hyperparameters, as described in Section IV-B4.

B. EXPERIMENTAL RESULTS

1) COMPARISON WITH OTHER METHODS

As shown in Table 4, MAMA surpassed the other methods in all phases, except for phase 6, in terms of the return rate. MAMA accomplished an average return rate of 19.1%, surpassing the second-best method, DASS+L, by 10.4%P. In particular, for phase 7 trending downwards, all methods experienced substantial losses except OLMAR and MAMA, which achieved positive returns. This was because MAMA selected a varying number of securities, determined their investment ratios, and determined the investment ratios among asset classes by considering concept drift and interest rates.

As demonstrated by the risk indicator in Tables 5 and 6, the average SR of MAMA was 1.71, surpassing that of the second-best method, DASS+L, by 0.76. The average MDD of MAMA was 0.11. For phase 6, MAMA exhibited a positive return rate; however, CORN performed the best. This is because CORN prioritized profit maximization rather than risk minimization, as exemplified by the average performances in terms of the SR and MDD. MAMA achieved more well-balanced profit and risk compared with CORN by integrating security selection, security allocation, and asset allocation as well as considering concept drift and interest rates.

Fig. 10 depicts the CAGR values of all methods over the entire test period. MAMA surpassed the other methods, achieving a CAGR of 40.9%, which was 23.0%P higher than that of the second-best method, DASS+L. This was attributed to the positive profits yielded by MAMA even in financial markets with high volatility, as in phases 3, 7, and 8, resulting in a compounded interest effect.

In summary, the results indicate that MAMA, which incorporates security selection, security allocation, and asset allocation, and combines all of the techniques listed in Table 1, is effective for portfolio management.

2) EFFECTIVENESS STUDY FOR ASSET ALLOCATION

Fig. 11 shows the allocations to cash made by MAMA over the entire test period. The base interest rate was the key policy rate set by a central bank to implement monetary policy [68]. We averaged the cash weight over non-overlapping two-month periods. During period A, MAMA gradually increased its allocation to cash because the price of SPY was perceived as overvalued owing to a long-term upward trend. During period B, MAMA rapidly increased its allocation to cash because the price of SPY exhibited a trend reversal from upward to downward. During period C, MAMA decreased its allocation to cash because the price of SPY was perceived as undervalued owing to a slowdown in its downward trend. During period D trending sideways, MAMA again increased its allocation to cash because of the increasing potential for financial market volatility as the base interest rate approached its peak. This increase during period D was similar to the insights presented by J.P. Morgan [69]. During period E, MAMA gradually decreased its allocation to cash as the price of SPY exhibited an upward trend. These results were observed because MAMA determined the investment ratios corresponding to different asset classes by combining the non-continual and continual learning at the asset class level.

3) ABLATION STUDIES

Ablation studies were conducted to evaluate the contribution of each MAMA component. To this end, behavior cloning (BC), asset allocation (AA), security selection (SS), security allocation (SA), continual learning (C), and non-continual

TABLE 4. Experimental results for each phase (return rate).

Phases	Return rate													
	CORN	OLMAR	All-Weather	EG	CRP	B&H	MASA-MLP	GRL	MASA-LSTM	GTAA-Agg3	MASA-DC	DASS	DASS+L	MAMA
1	-6.1%	3.7%	12.0%	7.7%	7.7%	7.8%	6.0%	9.5%	10.8%	5.7%	8.8%	1.9%	3.7%	**18.5%**
2	5.3%	3.5%	5.7%	4.7%	4.7%	3.8%	4.2%	6.0%	2.9%	7.5%	4.0%	3.0%	4.1%	**11.9%**
3	3.2%	1.8%	8.8%	-4.1%	-4.1%	-5.0%	-4.8%	-5.6%	8.1%	1.3%	8.1%	18.4%	20.4%	**27.7%**
4	15.3%	9.2%	6.8%	10.9%	10.9%	9.9%	11.4%	12.9%	10.5%	17.2%	18.1%	16.8%	13.0%	**26.5%**
5	-5.2%	18.5%	2.6%	8.9%	8.9%	11.0%	-4.4%	10.4%	14.6%	18.9%	5.1%	11.9%	9.9%	**19.8%**
6	**27.0%**	1.5%	5.7%	3.2%	3.2%	4.6%	5.7%	4.1%	2.4%	0.6%	2.1%	3.2%	6.5%	5.5%
7	-5.4%	5.1%	-13.8%	-3.9%	-3.9%	-1.7%	-11.5%	-3.7%	-11.1%	-0.8%	-8.2%	-2.3%	-1.0%	**7.1%**
8	-16.1%	-20.0%	-6.1%	0.4%	0.4%	-2.3%	8.5%	0.5%	-4.7%	-4.1%	3.0%	6.0%	7.5%	**9.8%**
9	-0.5%	3.2%	6.2%	2.3%	2.2%	3.1%	19.6%	1.9%	4.6%	6.4%	17.2%	7.3%	10.0%	**30.1%**
10	-4.6%	1.2%	3.0%	4.3%	4.3%	3.2%	4.1%	5.4%	3.6%	0.9%	5.7%	12.1%	13.2%	**34.2%**
Min.	-16.1%	-20.0%	-13.8%	-4.1%	-4.1%	-5.0%	-11.5%	-5.6%	-11.1%	-4.1%	-8.2%	-2.3%	-1.0%	**5.5%**
Max.	27.0%	18.5%	12.0%	10.9%	10.9%	11.0%	19.6%	12.9%	14.6%	18.9%	18.1%	18.4%	20.4%	**34.2%**
Avg.	1.3%	2.8%	3.1%	3.4%	3.4%	3.4%	3.9%	4.1%	4.2%	5.4%	6.4%	7.8%	8.7%	**19.1%**
Std.	0.12	0.10	0.08	0.05	0.05	**0.05**	0.09	0.06	0.08	0.08	0.08	0.07	0.06	0.10

TABLE 5. Experimental results for each phase (Sharpe ratio).

Phases	Sharpe ratio (SR)													
	CORN	OLMAR	All-Weather	EG	CRP	B&H	MASA-MLP	GRL	MASA-LSTM	GTAA-Agg3	MASA-DC	DASS	DASS+L	MAMA
1	-0.93	0.61	**5.00**	3.25	3.25	3.08	2.06	3.31	4.09	1.41	3.06	0.42	0.65	1.63
2	0.98	0.56	1.77	2.09	2.09	1.51	1.86	**2.19**	0.96	1.71	1.37	0.65	0.84	1.65
3	0.40	0.29	1.28	-0.35	-0.35	-0.47	-0.32	-0.39	1.03	0.24	1.03	1.05	1.03	**1.49**
4	1.55	1.15	1.85	2.99	**3.00**	2.87	1.65	2.95	2.52	2.14	2.30	1.87	1.49	1.59
5	-0.81	1.98	0.69	2.80	2.80	**3.12**	-0.52	2.73	2.18	1.95	1.25	1.65	1.35	2.42
6	**2.72**	0.25	1.66	1.01	1.01	1.36	1.90	1.08	0.85	0.15	0.58	0.52	0.46	0.63
7	-0.41	0.55	-2.44	-0.83	-0.83	-0.28	-2.33	-0.64	-1.97	-0.04	-1.53	-0.18	-0.03	**0.55**
8	-1.73	-1.77	-0.80	0.12	0.13	-0.27	1.34	0.15	-0.36	-0.96	0.56	0.55	0.79	**1.55**
9	0.02	0.46	1.31	0.78	0.77	0.79	2.85	0.57	0.96	1.38	**3.20**	0.97	0.99	2.79
10	-0.73	0.22	0.58	1.54	1.54	1.13	1.20	1.60	1.11	0.23	1.14	1.84	1.94	**2.80**
Min.	-1.73	-1.77	-2.44	-0.83	-0.83	-0.47	-2.33	-0.64	-1.97	-0.96	-1.53	-0.18	-0.03	**0.55**
Max.	2.72	1.98	**5.00**	3.25	3.25	3.12	2.85	3.31	4.09	2.14	3.20	1.87	1.94	2.80
Avg.	0.11	0.43	1.09	1.34	1.34	1.28	0.97	1.36	1.14	0.82	1.29	0.93	0.95	**1.71**
Std.	1.33	0.94	1.92	1.44	1.44	1.39	1.56	1.42	1.63	1.03	1.37	0.68	**0.55**	0.78

TABLE 6. Experimental results for each phase (maximum drawdown).

Phases	Maximum drawdown (MDD)													
	CORN	OLMAR	All-Weather	EG	CRP	B&H	MASA-MLP	GRL	MASA-LSTM	GTAA-Agg3	MASA-DC	DASS	DASS+L	MAMA
1	0.14	0.11	**0.01**	0.03	0.03	0.03	0.05	0.03	0.01	0.03	0.02	0.06	0.05	0.10
2	0.08	0.12	0.03	0.02	0.02	0.03	**0.01**	0.02	0.03	0.06	0.05	0.03	0.03	0.06
3	0.20	0.29	0.14	0.18	0.18	0.17	0.20	0.22	**0.13**	0.17	0.13	0.16	0.18	0.16
4	0.10	0.08	0.05	0.04	0.04	**0.04**	0.10	0.04	0.04	0.08	0.07	0.04	0.05	0.16
5	0.10	0.07	0.05	0.02	0.02	**0.02**	0.10	0.03	0.04	0.08	0.05	0.04	0.04	0.07
6	0.13	0.11	0.03	0.04	0.04	0.04	**0.02**	0.05	0.03	0.10	0.04	0.06	0.07	0.11
7	0.10	0.11	0.16	0.07	0.07	**0.06**	0.13	0.08	0.12	0.07	0.12	0.09	0.09	0.14
8	0.21	0.27	0.14	0.08	0.08	0.08	0.05	0.09	0.17	**0.05**	0.07	0.09	0.09	0.08
9	0.19	0.15	0.06	0.04	0.04	**0.04**	0.07	0.04	0.08	0.07	0.04	0.04	0.05	0.07
10	0.15	0.18	0.11	0.04	0.04	**0.04**	0.04	0.05	0.04	0.10	0.06	0.05	0.05	0.11
Min.	0.08	0.07	0.01	0.02	0.02	0.02	**0.01**	0.02	0.01	0.03	0.02	0.03	0.03	0.06
Max.	0.21	0.29	0.16	0.18	0.18	0.17	0.20	0.22	0.17	0.17	**0.13**	0.16	0.18	0.16
Avg.	0.14	0.15	0.08	0.06	0.06	**0.06**	0.08	0.07	0.07	0.08	0.07	0.07	0.07	0.11
Std.	0.05	0.08	0.05	0.05	0.05	**0.04**	0.06	0.06	0.05	0.04	0.04	0.04	0.04	0.04

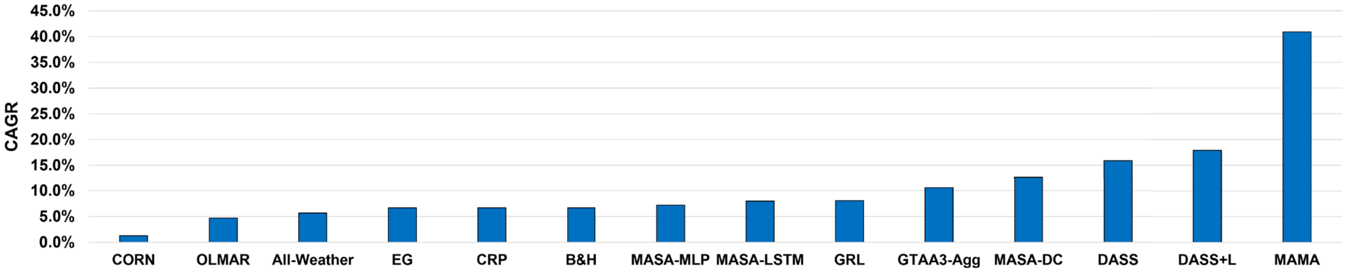


FIGURE 10. Comparison with other methods.

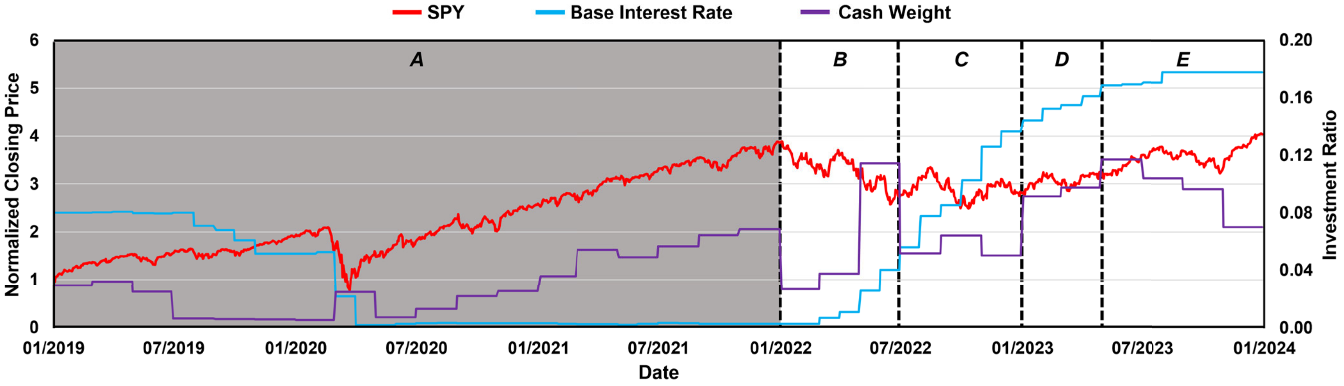


FIGURE 11. Investment ratio allocated to cash over the entire test period.

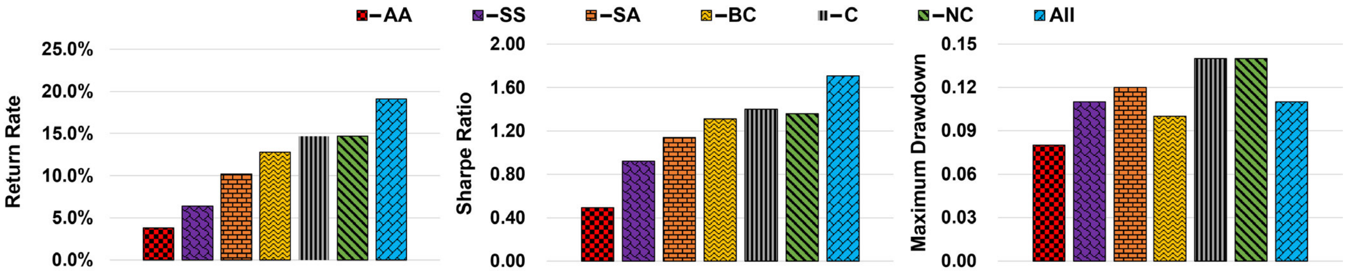


FIGURE 12. Experimental results of ablation studies.

learning (NC) were individually excluded. "All" represents MAMA with all components. To exclude the SS, we selected a fixed set of securities by skipping line 4 in Algorithm 1 and setting the top-K_A to 5. Fig. 12 depicts the average performances of the MAMA variants. The results show that all components contributed to the improvement of the performance, in the following order: AA, SS, SA, BC, C, and NC. In particular, the average performance was degraded when either the SS, AA, or SA was excluded. These results demonstrate that the components of MAMA complement one another.

4) COMPARISON WITH OTHER HYPERPARAMETER VALUES

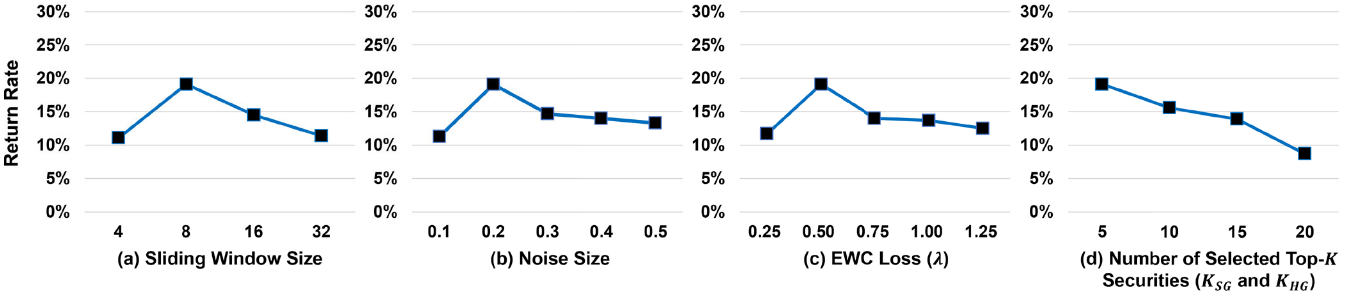
Fig. 13 shows the MAMA performance under different hyperparameter settings, including the sliding window size, noise size for exploration, λ in the EWC loss, and number of top-K_{SG} and K_{HG} securities selected from the ranking models of Algorithm 1. As illustrated in Fig. 13(a), the MAMA performance was degraded when the sliding window size was either very small or too large owing to insufficient or excessive information. Fig. 13(b) indicates that the MAMA performance was degraded when the noise size was either very small or large owing to the exploration-exploitation trade-off. Fig. 13(c) illustrates that the MAMA performance was degraded as λ increased because the continual inter-asset agent was adapted to volatilities in the financial market too slowly. Fig. 13(d) illustrates that the MAMA performance was degraded as K_{SG} and K_{HG} increased because securities with low profits were included in the portfolio.

5) ROBUSTNESS STUDY

Transaction costs, including transaction fees and taxes, are crucial in security trading. Hence, we verified the robustness of MAMA by adjusting the transaction cost rate ξ in Equations (3) and (11). Fig. 14 illustrates the average return rates under various transaction cost rates. For all methods, the average return rate decreased with an increasing transaction cost rate. Nevertheless, MAMA exhibited the best performance, even when the transaction cost rate was 0.3%, surpassing typical real-world trading costs. This is because MAMA generates a higher profit per trade than the transaction cost per trade.

6) RUNNING TIME COMPARISON

Table 7 represents the average training time and testing time per phase for MAMA and the second-best method (DASS+L). The experiments were conducted on a computer equipped with an Intel Core(TM) i7-7700 CPU at 3.60GHz, 16GB of RAM, and a Geforce RTX 3060 Ti Super GPU. As depicted in Table 7, the training and testing time of DASS+L were 2.24 and 2.15 faster than those of MAMA. This was because MAMA trained and tested security selection (SS), security allocation (SA), and asset allocation (AA) models for selecting securities and determining investment ratios across both securities and asset classes. In contrast,



(a) Sliding Window Size

(b) Noise Size

(c) EWC Loss (λ)

(d) Number of Selected Top-K Securities (K_{SG} and K_{HG})

FIGURE 13. Comparison with other hyperparameter values.

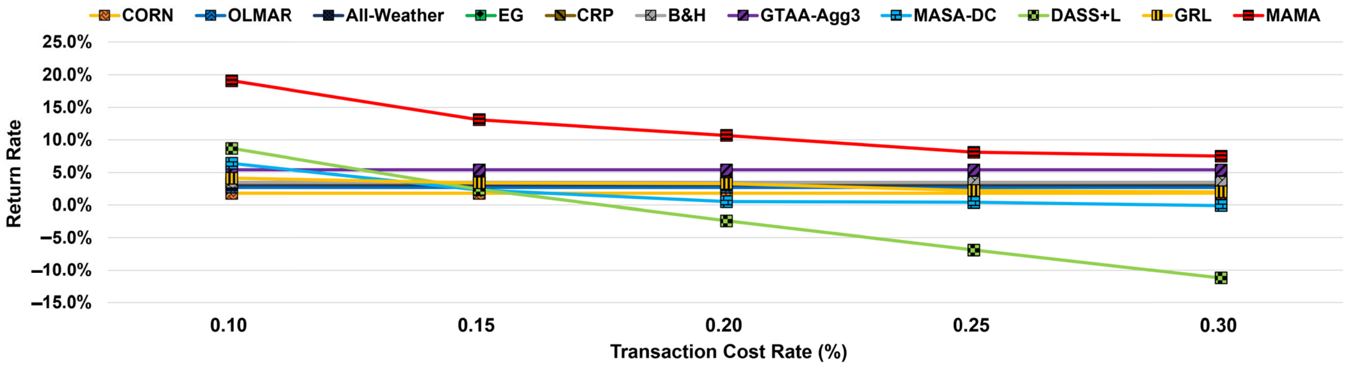


FIGURE 14. Experimental results of robustness study.

TABLE 7. Average running time per phase.

Methods	Training Time (secs)		Testing Time (secs)		Total Time (secs)
	SS	SA & AA	SS	SA & AA	
MAMA	601.00	2,090.10	1.01	1.40	2,693.51
DASS+L	1,197.25	-	1.12	-	1,198.37

TABLE 8. List of abbreviations.

Abbreviation	Description
CAGR	Compounded Annual Growth Rate
CE	Cross Entropy
EWC	Elastic Weight Consolidation
ETF	Exchange-Traded Fund
GNN	Graph Neural Network
GRU	Gated Recurrent Unit
HConv	Hypergraph Convolution
MAMA	Multi-Asset Multi-Agent reinforcement learning
MDP	Markov Decision Process
MSE	Mean Squared Error
SRL	State Representation Learning
TD3	Twin-Delayed Deep Deterministic policy gradient

DASS+L only trained and tested SS model for selecting securities. Within the architecture of MAMA, we employed GNN-based learning for the SS model and the RL algorithm for the SA and AA models. Compared to the SS model, the SA and AA models required approximately 3.48 times longer training time because they were optimized simultaneously through the complex RL algorithms that combine the techniques listed in Table 1, as described in Algorithms 2 and 3. Although the total time of MAMA was slower than that of DASS+L, it is less than one hour on a personal computer. Furthermore, MAMA surpassed DASS+L in terms of CAGR by 23.0%P.

V. CONCLUSION AND FUTURE WORK

We have proposed a novel portfolio management framework named MAMA that combines security selection, security allocation, and asset allocation as well as considers concept drift and interest rates to achieve fully autonomous and expert-level portfolio management. For security selection and allocation, we proposed a multi-agent structure in which each intra-asset agent learns the risk-return characteristics of its corresponding asset class. Each intra-asset agent is composed of a GNN-based SRL, which selects securities while adapting to concept drift at the security level, and an intra-asset RL, which determines the investment ratios among the selected securities. For asset allocation, we incorporated the non-continual and continual inter-asset agents to address concept drift at the asset class level. Each inter-asset agent is composed of an RNN-based SRL, which captures changes in interest rates by extracting temporal information from treasuries, and an inter-asset RL, which determines the investment ratio among asset classes. For portfolio construction and execution, we computed the investment ratios to construct the portfolio and implemented it through a trading system. We compared the performance of MAMA with those of state-of-the-art security selection, security allocation, asset allocation, and portfolio selection methods. The experimental results for stocks, bonds (including treasuries), and commodities revealed that MAMA achieves a CAGR of 40.9%, exceeding that of the second-best method by 23.0%P. In particular, MAMA achieves stable profits and mitigated risks by combining security selection, security allocation, and asset allocation. In future work, we plan to incorporate various asset classes, including real estate and cryptocurrency, and apply the framework to real-world investing scenarios.

VI. APPENDIX

See Table 8.

REFERENCES

[1] H. M. Markowitz and G. P. Todd, *Mean-Variance Analysis in Portfolio Choice and Capital Markets*, vol. 66. Hoboken, NJ, USA: Wiley, 2000.

[2] W. Junfeng, L. Yaoming, T. Wenqing, and C. Yun, "Portfolio management based on a reinforcement learning framework," *J. Forecasting*, vol. 43, no. 7, pp. 2792–2808, Nov. 2024.

[3] D.-Y. Park and K.-H. Lee, "Practical algorithmic trading using state representation learning and imitative reinforcement learning," *IEEE Access*, vol. 9, pp. 152310–152321, 2021.

[4] S.-H. Kim, D.-Y. Park, and K.-H. Lee, "Hybrid deep reinforcement learning for pairs trading," *Appl. Sci.*, vol. 12, no. 3, p. 944, Jan. 2022.

[5] A. Al Kuwaiti, K. Nazer, A. Al-Reedy, S. Al-Shehri, A. Al-Muhanna, A. V. Subbarayalu, D. Al Muhanna, and F. A. Al-Muhanna, "A review of the role of artificial intelligence in healthcare," *J. Personalized Med.*, vol. 13, no. 6, p. 951, 2023.

[6] S. Bahoo, M. Cucculelli, X. Goga, and J. Mondolo, "Artificial intelligence in finance: A comprehensive review through bibliometric and content analysis," *Social Netw. Bus. Econ.*, vol. 4, no. 2, p. 23, Jan. 2024.

[7] A. Chahal, S. R. Addula, A. Jain, P. Gulia, N. S. Gill, and V. B. Dhandayuthapani, "Systematic analysis based on conflux of machine learning and Internet of Things using bibliometric analysis," *J. Intell. Syst. Internet Things*, vol. 13, no. 1, pp. 196–224, 2024.

[8] M. T. Faber, "A quantitative approach to tactical asset allocation," *J. Wealth Manage.*, vol. 9, no. 4, pp. 69–79, Jan. 2007.

[9] W. J. Keller and J. W. Keuning, "Protective asset allocation (PAA): A simple momentum-based alternative for term deposits," *SSRN Electron. J.*, pp. 1–24, 2016. [Online]. Available: https://ssrn.com/abstract=2759734

[10] W. Keller and J. W. Keuning, "Breadth momentum and vigilant asset allocation (VAA): Winning more by losing less," *SSRN Electron. J.*, pp. 1–37, 2017. [Online]. Available: https://ssrn.com/abstract=3002624

[11] W. J. Keller and J. W. Keuning, "Breadth momentum and the Canary universe: Defensive asset allocation (DAA)," *SSRN Electron. J.*, pp. 1–29, 2018. [Online]. Available: https://ssrn.com/abstract=3212862

[12] T. Robbins, *Money Master the Game: 7 Simple Steps to Financial Freedom*. New York, NY, USA: Simon & Schuster, 2014. [Online]. Available: https://www.amazon.com/MONEY-Master-Game-Financial-Freedom/dp/1476757801

[13] R. Kim, C. H. So, M. Jeong, S. Lee, J. Kim, and J. Kang, "HATS: A hierarchical graph attention network for stock movement prediction," 2019, *arXiv:1908.07999*.

[14] F. Feng, X. He, X. Wang, C. Luo, Y. Liu, and T.-S. Chua, "Temporal relational ranking for stock prediction," *ACM Trans. Inf. Syst.*, vol. 37, no. 2, pp. 1–30, Apr. 2019.

[15] 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 *Proc. AAAI Conf. Artif. Intell.*, 2022, pp. 1112–1119.

[16] J. Lee, R. Kim, S.-W. Yi, and J. Kang, "MAPS: Multi-agent reinforcement learning-based portfolio management system," 2020, *arXiv:2007.05402*.

[17] R. Sawhney, S. Agarwal, A. Wadhwa, T. Derr, and R. R. Shah, "Stock selection via spatiotemporal hypergraph attention network: A learning to rank approach," in *Proc. AAAI Conf. Artif. Intell.*, 2021, vol. 35, no. 1, pp. 497–504.

[18] Y.-L. Hsu, Y.-C. Tsai, and C.-T. Li, "FinGAT: Financial graph attention networks for recommending top-K profitable stocks," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 1, pp. 469–481, Jan. 2023.

[19] S. Ha and F. J. Fabozzi, "Dual momentum: Testing the dual momentum strategy and implications for lifetime allocations," *J. Portfolio Manage.*, vol. 48, no. 4, pp. 282–301, Feb. 2022.

[20] J.-S. Kim, S.-H. Kim, and K.-H. Lee, "Portfolio management framework for autonomous stock selection and allocation," *IEEE Access*, vol. 10, pp. 133815–133827, 2022.

[21] S. Feng, C. Xu, Y. Zuo, G. Chen, F. Lin, and J. XiaHou, "Relation-aware dynamic attributed graph attention network for stocks recommendation," *Pattern Recognit.*, vol. 121, Jan. 2022, Art. no. 108119.

[22] Y. He, Q. Li, F. Wu, and J. Gao, "Static-dynamic graph neural network for stock recommendation," in *Proc. 34th Int. Conf. Sci. Stat. Database Manage.*, Jul. 2022, pp. 1–4.

[23] H. Wang, T. Wang, S. Li, J. Zheng, S. Guan, and W. Chen, "Adaptive long-short pattern transformer for stock investment selection," in *Proc. 31st Int. Joint Conf. Artif. Intell.*, Jul. 2022, pp. 3970–3977.

[24] X. Ma, T. Zhao, Q. Guo, X. Li, and C. Zhang, "Fuzzy hypergraph network for recommending top-K profitable stocks," *Inf. Sci.*, vol. 613, pp. 239–255, Oct. 2022.

[25] Q. Y. E. Lim, Q. Cao, and C. Quek, "Dynamic portfolio rebalancing through reinforcement learning," *Neural Comput. Appl.*, vol. 34, no. 9, pp. 7125–7139, May 2022.

[26] Z. Huang and F. Tanaka, "MSPM: A modularized and scalable multi-agent reinforcement learning-based system for financial portfolio management," *PLoS ONE*, vol. 17, no. 2, Feb. 2022, Art. no. e0263689.

[27] S. Shi, J. Li, G. Li, P. Pan, Q. Chen, and Q. Sun, "GPM: A graph convolutional network based reinforcement learning framework for portfolio management," *Neurocomputing*, vol. 498, pp. 14–27, Aug. 2022.

[28] Y.-C. Lin, C.-T. Chen, C.-Y. Sang, and S.-H. Huang, "Multiagent-based deep reinforcement learning for risk-shifting portfolio management," *Appl. Soft Comput.*, vol. 123, Jul. 2022, Art. no. 108894.

[29] T. T. Huynh, M. H. Nguyen, T. T. Nguyen, P. L. Nguyen, M. Weidlich, Q. V. H. Nguyen, and K. Aberer, "Efficient integration of multi-order dynamics and internal dynamics in stock movement prediction," in *Proc. 16th ACM Int. Conf. Web Search Data Mining*, Feb. 2023, pp. 850–858.

[30] G. Song, T. Zhao, S. Wang, H. Wang, and X. Li, "Stock ranking prediction using a graph aggregation network based on stock price and stock relationship information," *Inf. Sci.*, vol. 643, Sep. 2023, Art. no. 119236.

[31] H. Tian, X. Zhang, X. Zheng, and D. D. Zeng, "Learning dynamic dependencies with graph evolution recurrent unit for stock predictions," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 53, no. 11, pp. 6705–6717, Nov. 2023.

[32] J.-S. Kim, S.-H. Kim, and K.-H. Lee, "Diversified adaptive stock selection using continual graph learning and ensemble approach," *IEEE Access*, vol. 12, pp. 1039–1050, 2024.

[33] H. Zhang, Z. Shi, Y. Hu, W. Ding, E. E. Kuruoğlu, and X.-P. Zhang, "Optimizing trading strategies in quantitative markets using multi-agent reinforcement learning," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2024, pp. 136–140.

[34] K. Park, H.-G. Jung, T.-S. Eom, and S.-W. Lee, "Uncertainty-aware portfolio management with risk-sensitive multiagent network," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 1, pp. 362–375, Jan. 2024.

[35] C. Ma and S. Nan, "Dynamic graph reinforcement learning algorithm for portfolio management: A novel time–frequency correlated model," *Finance Res. Lett.*, vol. 63, May 2024, Art. no. 105373.

[36] Z. Li, V. H. Tam, and K. L. Yeung, "Developing a multi-agent and self-adaptive framework with deep reinforcement learning for dynamic portfolio risk management," in *Proc. AAMAS*, 2024, pp. 1174–1182.

[37] H. Sun, Y. Bian, L. Han, P. Zhu, D. Cheng, and Y. Liang, "Dynamic graph-based deep reinforcement learning with long and short-term relation modeling for portfolio optimization," in *Proc. 33rd ACM Int. Conf. Inf. Knowl. Manage.*, Oct. 2024, pp. 4898–4905.

[38] Q. Sun, X. Wei, and X. Yang, "GraphSAGE with deep reinforcement learning for financial portfolio optimization," *Expert Syst. Appl.*, vol. 238, Mar. 2024, Art. no. 122027.

[39] L.-C. Cheng and J.-S. Sun, ''Multiagent-based deep reinforcement learning framework for multi-asset adaptive trading and portfolio management,'' *Neurocomputing*, vol. 594, Aug. 2024, Art. no. 127800.

[40] D. W. Jeong, S. J. Yoo, and Y. H. Gu, ''Safety AARL: Weight adjustment for reinforcement-learning-based safety dynamic asset allocation strategies,'' *Expert Syst. Appl.*, vol. 227, Oct. 2023, Art. no. 120297.

[41] R. Jammazi, R. Ferrer, F. Jareño, and S. M. Hammoudeh, ''Main driving factors of the interest rate-stock market Granger causality,'' *Int. Rev. Financial Anal.*, vol. 52, pp. 260–280, Jul. 2017.

[42] J. Zhao and Y. Han, ''Interest rate fluctuations and corporate financial leverage,'' *Finance Res. Lett.*, vol. 80, Jun. 2025, Art. no. 107344.

[43] J. Wang, G. Song, Y. Wu, and L. Wang, ''Streaming graph neural networks via continual learning,'' in *Proc. 29th ACM Int. Conf. Inf. Knowl. Manage.*, Oct. 2020, pp. 1515–1524.

[44] W. L. Hamilton, R. Ying, and J. Leskovec, ''Inductive representation learning on large graphs,'' in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2022, pp. 1024–1034.

[45] Y. Jiang, J. Olmo, and M. Atwi, ''Deep reinforcement learning for portfolio selection,'' *Global Finance J.*, vol. 62, Sep. 2024, Art. no. 101016.

[46] T. Kum, E. Koay, D. Maskell, and C. Quek, ''Portfolio rebalancing using deep reinforcement learning,'' in *Proc. 11th Int. Conf. Comput. Artif. Intell. (ICCAI)*, Mar. 2025, pp. 718–726.

[47] E. Huang and Y. Lawryshyn, ''Deep hedging under market frictions: A comparison of DRL models for options hedging with impact and transaction costs,'' *J. Risk Financial Manage.*, vol. 18, no. 9, p. 497, Sep. 2025.

[48] P. Ghosh, A. Neufeld, and J. K. Sahoo, ''Forecasting directional movements of stock prices for intraday trading using LSTM and random forests,'' *Finance Res. Lett.*, vol. 46, May 2022, Art. no. 102280.

[49] Y.-S. Jeong, M. K. Jeong, and O. A. Omitaomu, ''Weighted dynamic time warping for time series classification,'' *Pattern Recognit.*, vol. 44, no. 9, pp. 2231–2240, Sep. 2011.

[50] L. E. Peterson, ''K-nearest neighbor,'' *Scholarpedia*, vol. 4, no. 2, p. 1883, 2009.

[51] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lió, and Y. Bengio, ''Graph attention networks,'' in *Proc. ICLR*, 2020, pp. 1–12.

[52] S. Bai, F. Zhang, and P. H. S. Torr, ''Hypergraph convolution and hypergraph attention,'' *Pattern Recognit.*, vol. 110, Feb. 2021, Art. no. 107637.

[53] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska, D. Hassabis, C. Clopath, D. Kumaran, and R. Hadsell, ''Overcoming catastrophic forgetting in neural networks,'' *Proc. Nat. Acad. Sci. USA*, vol. 114, no. 13, pp. 3521–3526, 2017.

[54] T. M. Cover, ''Universal portfolios,'' *Math. Finance*, vol. 1, no. 1, pp. 1–29, Jan. 1991.

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

[56] A. Borodin, R. El-Yaniv, and V. Gogan, ''Can we learn to beat the best stock,'' in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 16, 2003, pp. 1–9.

[57] 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, May 2012.

[58] B. Li and S. C. H. Hoi, ''On-line portfolio selection with moving average reversion,'' 2012, *arXiv:1206.4626*.

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

[60] L. Györfi, G. Lugosi, and F. Udina, ''Nonparametric kernel-based sequential investment strategies,'' *Math. Finance*, vol. 16, no. 2, pp. 337–357, Apr. 2006.

[61] B. Li, S. C. Hoi, and V. Gopalkrishnan, ''CORN: Correlation-driven nonparametric learning approach for portfolio selection,'' *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, pp. 1–29, 2011.

[62] *ETF Database Categories*. Accessed: May 26, 2025. [Online]. Available: https://etfdb.com/etfdb-categories/

[63] *Yahoo Finance*. Accessed: May 26, 2025. [Online]. Available: https://finance.yahoo.com/

[64] G. Chakravorty, A. Awasthi, and B. Da Silva, ''Deep learning for global tactical asset allocation,'' *SSRN J.*, pp. 1–17, 2018. [Online]. Available: https://ssrn.com/abstract=3242432

[65] W. F. Sharpe, ''The Sharpe ratio,'' *J. Portfolio Manage.*, vol. 21, no. 1, pp. 49–58, Oct. 1994.

[66] D. P. Kingma and J. Ba, ''Adam: A method for stochastic optimization,'' 2014, *arXiv:1412.6980.*

[67] A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada, ''Control barrier functions: Theory and applications,'' in *Proc. 18th Eur. Control Conf. (ECC)*, Jun. 2019, pp. 3420–3431.

[68] M. Alexandre and G. T. Lima, ''Combining monetary policy and prudential regulation: An agent-based modeling approach,'' *J. Econ. Interact. Coordination*, vol. 15, no. 2, pp. 385–411, Apr. 2020.

[69] *Global Asset Allocation Views*. Accessed: May 26, 2025. [Online]. Available: https://am.jpmorgan.com/content/dam/jpm-am-aem/americas/br/en/insights/portfolio-insights/global-asset-allocation-views-br-en.pdf

SANG-HO KIM received the B.S. degree in computer engineering from Kwangwoon University, Seoul, Republic of Korea, in 2023, where he is currently pursuing the integrated M.S. and Ph.D. degree in computer engineering.

KI-HOON LEE received the B.S., M.S., and Ph.D. degrees in computer science from Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Republic of Korea, in 2000, 2002, and 2009, respectively.

From 2010 to 2012, he was a Manager with the Advanced Institute of Technology, Korea Telecom (KT). From 2012 to 2013, he was a Senior Developer with SAP Labs Korea. He joined Kwangwoon University, in 2013, where he is currently a Professor with the School of Computer and Information Engineering. He has published papers in leading international journals and conferences, including IEEE Access, *VLDB Journal*, *SIGMOD Record*, and IEEE ICDE.