BDM: Exploring Certain Profitable Opportunities in Uncertain Financial Trading Environments

Abstract

This paper presents BDM, a behavior-diversified multi-agent trading algorithm developed to tackle the challenges of non-stationarity and "zero-sum" dynamics in uncertain financial trading environments. Existing multi-agent methods often underestimate the importance of integrating behavior diversity, action sparsity, and non-trading-period optimization, all of which are crucial for achieving stable and high trading performance. To overcome these limitations, we introduce hedge sampling methods for constructing price trends and augmented technical indicators, enabling the training of deep reinforcement learning agents with distinct yet complementary trading behaviors. Additionally, we address action sparsity through a combination of agent-switching and action-sparsification strategies. By optimizing non-trading periods, BDM effectively explores profitable opportunities, even in uncertain market situations with high transaction costs. Notably, with transaction costs set at 0.1%, BDM outperforms eleven representative algorithms on five out of six real-world datasets. BDM achieves a cumulative wealth and Calmar ratio that are 1.49 and 1.5 times higher, respectively, compared to the buy-and-hold strategy.

1 INTRODUCTION

Achieving stable returns by trading a set of assets in uncertain financial markets presents a significant challenge in studying uncertainty within artificial intelligence [Ye et al., 2020, Wang et al., 2021, Yin et al., 2021, Huang et al., 2024, Shavandi and Khedmati, 2022, Huang and Tanaka, 2022]. Unlike domains like natural language processing (NLP) and computer vision (CV), where intelligent agents have made remarkable advancements, the real-world financial trading environment exhibits distinctive characteristics, including non-stationarity, zero-sum dynamics, and significant operational costs such as transaction costs [Ding et al., 2021, Ofer et al., 2021]. These unique features contribute to the increased complexity of developing automated agents compared to those in the NLP and CV domains.

The non-stationary nature of volatile financial trading environments becomes apparent through the observed price time series. These series can display a range of patterns, including trends, cycles, random walks, or combinations thereof. Such non-stationary data presents a challenge for general multi-period learning methods [Ye et al., 2020, Zhang et al., 2020b], including deep reinforcement learning (DRL), which heavily rely on the Markov process assumption and are limited in their ability to quickly respond to the volatile changes in short-term market trends [Niu et al., 2022].

In addition, the zero-sum game characteristic of financial markets, where gains for one participant come at the expense of others [Roberts and Davidai, 2022], leads to intense competition among trading agents for limited capital resources. Consequently, any trading strategy that proves effective in previous periods becomes ineffective in subsequent periods. This is due to the fact that any change in price signals is immediately observable to other trading agents through market price signals.

This rapidly evolving and dynamic financial trading environment underscores the significance of studying portfolio diversity, as demonstrated in studies conducted by Cover and others [Cover, 1991, Lai et al., 2018, He and Yang, 2022, Yang et al., 2020b, 2022]. In this context, portfolio algorithms are regarded as agents. By harnessing the collective intelligence exhibited by these agents, portfolio ensemble algorithms have the potential to mitigate downside risk and enhance trading robustness. However, it is important to note that existing trading ensemble algorithms often overlook three crucial aspects: behavior diversity, action sparsity and on-trading-period optimization.

Submitted to the 40th Conference on Uncertainty in Artificial Intelligence (UAI 2024). **To be used for reviewing only**.

The behavior-diversity of portfolio ensemble algorithms relies on having a pool of agents capable of exhibiting a variety of trading behaviors [Lee et al., 2021]. However, this requirement is often overlooked by many algorithms [Yang et al., 2020b, 2022, Fior and Cagliero, 2022, Kabbani and Duman, 2022, Zhang et al., 2022b, Huang et al., 2024, Shavandi and Khedmati, 2022, Huang and Tanaka, 2022], leading to a lack of diversified behaviors within the agent pool.

Sparse portfolios optimize returns by strategically allocating capital to a carefully selected subset of assets for future trading. Trading algorithms commonly incorporate constraints like vector norms or set cardinality in their optimization objectives, as discussed in Li et al. [2022], Wang et al. [2023], Hao et al. [2021]. However, these algorithms have limitations. Firstly, sparse constraints, such as the l_1 norm, may not adequately capture the diversity of assets [Aquino et al., 2023], including those identified by effective technical indicators for price prediction [Shynkevich et al., 2017, Ji et al., 2022]. Secondly, combining sparsity and diversity objectives in a single optimization objective poses challenges [Chen and Zhou, 2022].

In real-world trading scenarios, excessive trading can result in significant loss of returns. As a result, many algorithms incorporate transaction costs as a penalty term to constrain a single optimization objective Zhang et al. [2020b], Guo et al. [2023]. However, this approach presents challenges in achieving a balance between the increasing optimization requirements, including behavior diversity and action sparsity, within a single optimization objective.

To address behavior-diversity and action-sparsity while considering high transaction costs, we introduce the behavior-diversified multi-agent trading algorithm (BDM) [1]. BDM offers an effective solution for generating optimal investment actions by utilizing a pool of agents with diverse hedged trading behaviors. The key contributions of our work can be summarized as follows:

- By training deep reinforcement learning (DRL) agents using a novel feature construction, including short-term hedged and diverse price trends, we establish a behavior-diverse agent pool. This approach allows us to consistently identify profitable opportunities by leveraging the diversified return profiles exhibited by our agent pool.
- To achieve action sparsity and optimize trading performance, we propose a combination of agent-switching and action-sparsification strategies. This distills the trading knowledge of our agent pool, enhancing its effectiveness.
- We introduce an adaptive method for determining optimal non-trading periods, which reduces the costs associated with frequent trading operations.

In summary, by integrating these key design points, BDM demonstrates robust capabilities in effectively identifying and capturing profit opportunities amidst the uncertainties of the financial trading landscape.

2 RELATED WORK

This section presents an overview of the related work on methods that employ an ensemble of agents and action sparsification for portfolio management in uncertain financial environments.

2.1 ENSEMBLE OF TRADING AGENTS

The use of multiple trading agents for sequential decision-making has gained popularity to enhance portfolio diversification in uncertain trading environments [Chen et al., 2023, Yang et al., 2020a, Lee et al., 2021, Riva et al., 2022, Huang et al., 2024]. For example, Chen et al. [2023] introduced a role-aware multi-agent algorithm that categorizes trading agents into distinct groups, providing diverse observation sets and reward functions to simulate real-world investment behaviors. Shavandi and Khedmati [2022] trained agents using deep Q-Networks (DQN) across diverse timeframes. However, relying solely on price information from different time intervals limits the exploitation of hedging opportunities and diverse price information.

Switching between multiple agents provides an effective alternative to relying solely on weighted averaging of agent advice [Li et al., 2017, Guo and Ching, 2021]. For example, the binary switch portfolio (BSP) algorithm [Li et al., 2017] dynamically transitions between the successive constant rebalanced portfolio (SCRP) and the buy and hold (BAH) portfolio using a simple binary classification rule. Yang et al. [2020a] integrated the proximal policy optimization (PPO), advantage actor-critic (A2C), and the deep deterministic policy gradient (DDPG) algorithms to develop a more robust and adaptable trading strategy. However, a common drawback of these ensemble methods resides in the lack of behavior diversity in constructing the agent pool.

2.2 SPARSE INVESTMENT DECISIONS

The sparse portfolio strategy aims to enhance returns by concentrating capital in a carefully selected subset of assets [Lee et al., 2018, Li et al., 2022, Wang et al., 2023]. For example, Lee et al. [2018] proposed a sparse Markov decision process (MDP) with a novel causal sparse Tsallis entropy regularization. Another approach byLi et al. [2022] minimized tracking error while explicitly controlling the number of assets selected for the portfolio. However, many

[1]The code is available at https://github.com/ccckkkyyy666/BDM

deep reinforcement learning (DRL) based portfolio algorithms often overlook the importance of generating sparse actions. For instance,Zhang et al. [2021] introduced a novel portfolio management strategy within the DDPG framework, whileLin and Beling [2021] presented an end-to-end PPO framework utilizing a sparse reward function.

2.3 NON-TRADING-PERIOD OPTIMIZATION

High-frequency trading in real-world scenarios can lead to substantial losses in returns. To address this, algorithms in financial trading literature incorporate transaction costs as a penalty term in the optimization objective, aiming to balance returns and minimize changes in adjacent portfolio vectors [Zhang et al., 2020b, Guo et al., 2023]. For example, Zhang et al. [2020b] proposed a cost-sensitive portfolio selection method, while Guo et al. [2023] developed a net profit maximization model considering transaction costs. However, incorporating transaction costs as a penalty term poses challenges in achieving an optimal balance between optimization requirements like behavior diversity and sparse portfolios.

Another approach to mitigate high transaction costs is to pause trading periods periodically. By implementing non-trading periods with a buy-and-hold strategy, algorithms can reduce trading operation costs. Determining adaptive non-trading periods is crucial for optimizing this strategy. Huang et al. [2015] demonstrated the effectiveness of this approach. By strategically timing these pauses, algorithms can manage transaction costs and optimize portfolio performance.

3 TRAINING OF THE AGENT POOL

This section presents the methodology used to train a pool of agents, enabling them to exhibit hedge trading behaviors in various financial market situations.

3.1 HEDGED PRICE-TREND FEATURES

The agent pool consists of five types of agents: $S = \{M_{max}, M_{min}, M_{mean}, M_{ema}, M_{real}\}$. To train each of these agent types, we design five corresponding price-trend features, which are defined as follows:

$$\begin{aligned}
\boldsymbol{p}_t^{max} &= \max_{0\le k\le w-1} \boldsymbol{p}_{t-k}, \text{ for } M_{max},\\
\boldsymbol{p}_t^{min} &= \min_{0\le k\le w-1} \boldsymbol{p}_{t-k}, \text{ for } M_{min},\\
\boldsymbol{p}_t^{mean} &= \frac{1}{w}\sum_{k=0}^{w-1} \boldsymbol{p}_{t-k}, \text{ for } M_{mean},\\
\boldsymbol{p}_t^{ema} &= \sum_{k=0}^{w-1} \beta(1-\beta)^k \boldsymbol{p}_{t-k}, \text{ for } M_{ema},\\
\boldsymbol{p}_t^{real} &= \boldsymbol{p}_t, \text{ for } M_{real}.
\end{aligned} \tag{1}$$

In this context, the input price vector $\boldsymbol{p}_t \in \mathbb{R}_+^m$ can represent open prices, high prices, low prices, or close prices. The parameter w corresponds to the recent time window.

To foster diverse trading behaviors within a pool of agents, particularly in volatile market situations, we employ two sampling methods for price-trend feature construction. The first method utilizes single-sample sampling with expressions for $\boldsymbol{p}_t^{max}$ and $\boldsymbol{p}_t^{min}$ in Eq. (1). This method effectively captures oversold or overbought market sentiments and has demonstrated strong performance in previous studies [Lai et al., 2017, Dai et al., 2022]. The second method involves multiple-sample sampling with expressions for $\boldsymbol{p}_t^{mean}$ and $\boldsymbol{p}_t^{ema}$ in Eq. (1). It focuses on smoothing price trends over a recent time window [Li et al., 2015].

In addition to the four price features $\boldsymbol{p}_t^{max}$, $\boldsymbol{p}_t^{min}$, $\boldsymbol{p}_t^{mean}$, and $\boldsymbol{p}_t^{ema}$, we include the original input price feature $\boldsymbol{p}_t^{real}$ as a fundamental source of information within the hedge structure of price features.

3.2 STATE AND ACTION SPACES OF AGENTS

The states $\boldsymbol{s}_t^i$ in the state space serve as inputs for training the i-th agent to generate actions. Each state $\boldsymbol{s}_t^i$ is defined as follows:

$$\boldsymbol{s}_t^i = \text{vec}(cash_k, \boldsymbol{p}_k^i, \boldsymbol{b}_k, \mathbf{ind}_k)_{k=t-w}^t. \tag{2}$$

The symbol $cash_k \in \mathbb{R}_+$ represents the remaining cash at period k. The symbol $\boldsymbol{p}_k^i$ denotes the price-trend features used by the i-th type of agents, where $i \in \{max, min, mean, ema, real\}$ as defined in Eq. (1). $\boldsymbol{b}_k$ denotes the shares of m assets held by an agent, and $\mathbf{ind}_k$ represents the vector consisting of eight technical indicators, as described in Table 1. The operator vec vertically stacks the input vectors $cash_k$, $\boldsymbol{p}_k$, $\boldsymbol{b}_k$, and $\mathbf{ind}_k$ into form a state $\boldsymbol{s}_t^i$.

Table 1: Technical indicators used for augmented features

Indicator	Symbol	Discription
macd	M	Moving Average Convergence Divergence
boll_ub	B_u	Bollinger Bands Upper Band
boll_lb	B_l	Bollinger Bands Lower Band
rsi_30	R	Relative Strength Index for 30 periods
cci_30	C	Commodity Channel Index for 30 periods
dx_30	DX	Directional Movement Index for 30 periods
ma_30	M_{30}	30-Period Simple Moving Average of Closing Prices
ma_60	M_{60}	60-Period Simple Moving Average of Closing Prices

The agents' action space consists of allowable actions $\boldsymbol{a}_{t+1} \in \mathbb{Z}^m$ for m assets in a given state $\boldsymbol{s}_t^i$, where m is set to 29. This deliberate selection of 29 assets effectively captures core situations in their respective financial markets, ensuring capital efficiency. Actions can involve buying ($a_{t+1,i} > 0$), selling ($a_{t+1,i} < 0$), or holding ($a_{t+1,i} = 0$) assets. For instance, $\boldsymbol{a}_{t+1} = (10, -10, 0, ...)$ indicates buy-

ing 10 shares of the first asset, selling 10 shares of the second asset, and holding the remaining shares of the third asset.

3.3 TRAINING OBJECTIVE

All agents in the pool are trained based on the proximal policy optimization (PPO) [Schulman et al., 2017, Hsu et al., 2020]. The objective function is defined as follows:

$$\max L_p + c_e L_e + c_v L_v, \tag{3}$$

where c_e and c_v, utilized for scaling, are set to 0.01 and 0.5, respectively.

The first component in Eq. (3) is the policy loss L_p, which is defined as follows:

$$L_p = \min\left\{\rho_t(\theta)\hat{A}_t, \text{clip}(\rho_t(\theta),\ 1-\epsilon,\ 1+\epsilon)\hat{A}_t\right\}, \tag{4}$$

where, ϵ is a hyperparameter set to 0.2 [Liu et al., 2021], and $\rho_t(\theta) = \pi_\theta(a_t|s_t)/\pi_{\theta_{old}}(a_t|s_t)$ denotes the probability ratio at period t. The scalar $\hat{A}_t$ in Eq. (4) serves as an estimator of the advantage function and is defined as follows:

$$\hat{A}_t = \delta_t + (\gamma\lambda)\delta_{t+1} + ... + (\gamma\lambda)^{T-t+1}\delta_{T-1}. \tag{5}$$

The discount factors γ and λ are set to γ to 0.99 and λ to 1, respectively. The scalar δ_t is defined by $\delta_t = r_t + \gamma V_{t+1} - V_t$, where r_t represents the reward:

$$r_t = \eta\left(\omega_t - \omega_{t-1}\right),$$

where the symbol ω_t represents the cumulative wealth (CW) obtained by an agent at period t as defined in Eq. (9), and the scaling factor η is set to a default value of $1E-04$.

The second component in Eq. (3) is the entropy loss L_e. The introduction of L_e enables the policy to maintain a certain degree of randomness when generating actions, as defined as follows:

$$L_e = -H[-\log p(\boldsymbol{a}_t)],$$

where $p(\boldsymbol{a}_t)$ denotes the probability of taking the action $\boldsymbol{a}_t$.

The third component in Eq. (3) is the value loss L_v, which quantifies the difference between the predicted value from the policy evaluation network and the actual return using the mean square error:

$$L_v = ||r_t - (V_{t-1} + \text{clip}(V_t - V_{t-1}, -\epsilon_{\text{vf}},\ \epsilon_{\text{vf}}))\,||^2,$$

where the symbol ϵ_{vf} represents the parameter used for clipping the advantage of values V_t and V_{t-1}.

4 THE INFERENCE BASED ON THE AGENT POOL

This section provides a detailed explanation of the key components of the BDM algorithm. These components work collaboratively within the inference architecture of BDM to generate investment actions from the ensemble of trained agents.

4.1 INFERENCE ARCHITECTURE

The key components of the BDM algorithm consist of the pool of S trained agents, agent selection, and action sparsification. The inference architecture depicted in Figure 1 illustrates the data flows between these components. There are two distinct types of trading environments: individual trading environments utilized by each agent within the agent pool and a switch environment employed by the agent selection process. These trading environments serve as repositories for storing trading state information, such as the number of shares $\boldsymbol{b}_t$ held and the current cumulative wealth ω_t, as defined in Eq. (9).

4.2 AGENT SELECTION

This section introduces a method to select effective agents within the agent pool S. Firstly, we propose a performance vector $\boldsymbol{c}_t \in \mathbb{R}^{|S|}$ to evaluate each agent. The $\boldsymbol{c}_t$ is defined as follows:

$$\boldsymbol{c}_t = \alpha\frac{\boldsymbol{\omega}_t - \boldsymbol{\omega}_{t-5}}{5} + (1-\alpha)(\boldsymbol{\omega}_t - \boldsymbol{\omega}_c), \tag{6}$$

where the first term captures the short-term change of cumulative wealth (CW), and the second term represents the long-term change of CW. The symbol $\boldsymbol{\omega}_c \in \mathbb{R}_+^{|S|}$ denotes the previous CW vector at the switching period $c \in [1, t]$.

Secondly, based on the definition of $\boldsymbol{c}_t$, we select the agent indexed by $i = \arg\max_i c_{t,i}$ from the agent pool S.

4.3 ACTION SPARSIFICATION

Merely selecting an agent from the agent pool is insufficient to ensure sparse investment decisions. This section introduces a method to generate sparse investment actions based on the corresponding actions of the selected agents.

Let $i = \arg\max_i a_{t+1,i}$ denote the asset with the highest weight in the action $\boldsymbol{a}_{t+1}$. To satisfy the sparsity condition for holding shares, we update $\boldsymbol{a}_{t+1}$ to $\boldsymbol{a}'_{t+1}$ using the following formula:

$$\begin{cases} a'_{t,j} = -b_{t,j},\ \forall j \neq i \wedge b_{t,j} > 0, \\ a'_{t,i} = \frac{1}{p_{t,i}}\left(cash_t + \sum_{j\neq i} p_{t,j}b_{t,j}\right), \end{cases} \tag{7}$$

where $p_{t,i}$ represents the price of the ith asset, $b_{t,j} \in \mathbb{R}_+$ denotes the number of shares of the jth asset, and $cash_t$ denotes the available cash at period t.

The term $\sum_{j\neq i} p_{t,j}b_{t,j}$ represents the capital obtained from selling assets other than the ith asset, which is achieved by setting the actions for assets j as $a'_{t,j} = -b_{t,j}$.

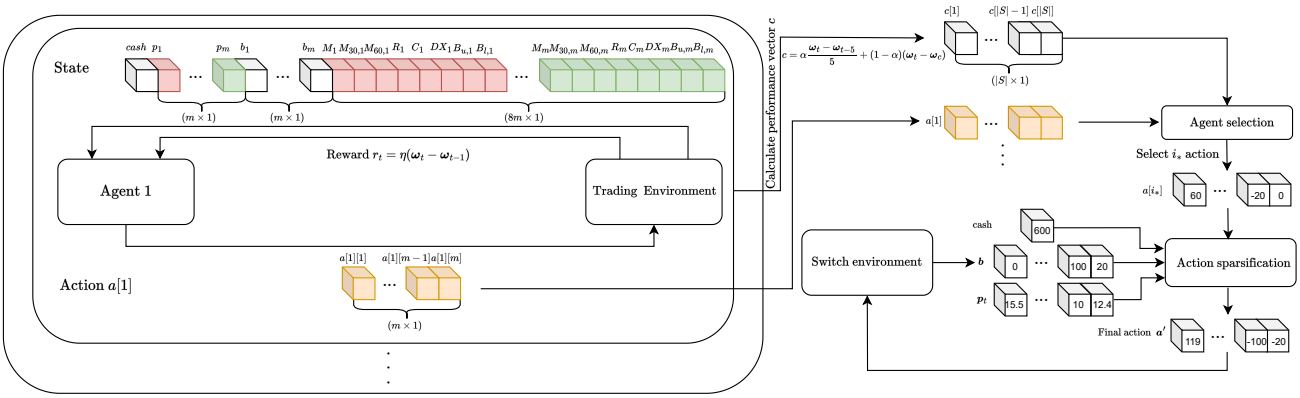


Figure 1: The inference architecture of the BDM algorithm.

4.4 OPTIMAL SETTING OF NON-TRADING PERIODS

In a real-world trading environment, transaction fees cannot be overlooked. Particularly, excessive trading can significantly reduce the cumulative wealth of the ensemble of agents. Through extensive testing on real-world datasets, we have discovered that the optimal number of non-trading periods at period t, denoted as d_t^*, can be determined using the following objective:

$$d_t^* = \arg\max_{d \in D} \mathbf{1}^\intercal \left(\boldsymbol{\omega}_t - \boldsymbol{\omega}_{t-d} \right). \tag{8}$$

In this case, the candidate set of non-trading periods is represented by $D = \{2, 5, 7\}$, where the longest non-trading period can extend up to a week. Eq. (8) employs a criterion where the optimal non-trading period d_t^* can result in the maximum increase in the ensemble wealth $\boldsymbol{\omega}_t$ over a span of d_t^* periods.

4.5 THE BDM ALGORITHM

This section presents the behavior-diversified multi-Agent trading (BDM) algorithm, described in Algorithm 1. The algorithm is implemented using the DRL library [2]. In the BDM algorithm, two types of trading environment objects are initialized: switchEnv and env$[k]$, where $k \in [1, S]$ denotes the agent index. These objects store the state information for the switching agent and the agents in the agent pool, respectively. To access the remaining capital at time t, one can retrieve the value from the $\mathrm{asset\_memory}[t]$ property of a trading environment object.

The time complexity of the BDM algorithm is $O(n(|S|\xi + m))$, with ξ representing the inference time of the neural network for a DRL agent, n being the number of periods, and m indicating the number of assets. The most significant time consumption occurs during Step 12 and Step 13, where neural network inference is performed for each of the $|S|$ agents.

5 EXPERIMENT RESULTS

We conducted comprehensive experiments on six diverse and publicly available datasets from real-world markets. The experiments were performed on a machine equipped with an AMD Ryzen 7 5800H processor, an NVIDIA GeForce RTX 3060, and 16GB RAM.

5.1 DATASETS

The datasets utilized in this study were sourced from the Yahoo Finance API, encompassing a diverse range of financial trading environments. These datasets include the Dow Jones Industrial Average Stock Index (DOW), Shanghai and Shenzhen Indexes (HS), Hong Kong Index (HK), Financial Times Stock Exchange Index (FTSE), New York Stock Exchange Index (NYSE), along with global cryptocurrency market data (CRYPTO). The overview of each dataset can be found in Table 2. To assess the generalization ability of our agents, the test periods are sampled independently from the train periods, ensuring they are not continuous. As described in Section 3.2, the deep neural networks employed to train our agents are specifically tailored to generate actions for investing in a predetermined set of 29 assets. With this in mind, we randomly sample 29 assets from the corre-

Table 2: Overview of the datasets

Dataset	Region	Train Periods	Test Periods	# Assets
DOW	US	02/07/2010 - 01/07/2011	02/01/2013 - 02/01/2015	29
HS	CN	05/01/2011 - 05/01/2012	05/01/2015 - 30/12/2017	29
CRYPTO	Global	07/10/2020 - 07/10/2021	28/09/2022 - 25/06/2023	29
HK	CN	27/11/2019 - 27/11/2020	04/01/2021 - 04/01/2022	29
NYSE	US	04/02/2013 - 04/02/2014	02/10/2017 - 01/10/2018	29
FTSE	UK	03/02/2019 - 03/02/2020	30/06/2022 - 30/06/2023	29

[2] https://github.com/AI4Finance-Foundation/FinRL

Algorithm 1 BDM: behavior-diversified multi-agent trading algorithm

Input:

Agent pool: $S = \{M_{max}, M_{min}, M_{mean}, M_{ema}, M_{real}\}$,
Agent switching coefficient: $\alpha = 0.5$,
Non-trading periods: $D = \{2, 5, 7\}$,
Number of assets: m
Agent environments: StockTradingEnv, switchEnv.

Procedure:

Initialize $\omega = 1,000,000$, $d_* = 2$, $i_c = 1$, $\boldsymbol{\omega}_c = \mathbf{0}$
for $k = 1,...,|S|$ **do**
  env$[k]$, obs$[k]$ = StockTradingEnv(ω).get_sb_env()
end for
for $t = 1 \rightarrow n$ **do**
  for $k = 1,...,|S|$ **do**
    $\boldsymbol{a}[k] = S[k]$.predict(obs$[k]$)$[0]$
    obs$[k]$ = env$[k]$.step$(a[k])[0]$
  end for
  $\boldsymbol{a}' = \boldsymbol{a}[1]$
  if $t < \max(D)$ **then**
    switchEnv.step$(\boldsymbol{a}')$
    continue
  end if
  $\boldsymbol{\omega}_t$ = [env[1].asset_memory$[t]$, ..., env$[|S|]$.asset_memory$[t]]^\intercal$
  $\boldsymbol{\omega}_{t-5}$ = [env[1].asset_memory$[t-5]$, ..., env$[|S|]$.asset_memory$[t-5]]^\intercal$
  $\boldsymbol{c}_t = \alpha \frac{\boldsymbol{\omega}_t - \boldsymbol{\omega}_{t-5}}{5} + (1-\alpha)(\boldsymbol{\omega}_t - \boldsymbol{\omega}_c)$
  $i_* = \arg\max_i s_{t,i}$ //select the best agent i_*.
  $\boldsymbol{a}' = \mathbf{0}$
  if $t \,\%\, d_* == 0$ **then**
    $d_* = \arg\max_{d \in D} \mathbf{1}^\intercal(\boldsymbol{\omega}_t - \boldsymbol{\omega}_{t-d})$
    if $i_c \neq i_*$ **then**
      $\boldsymbol{\omega_c} = \boldsymbol{\omega_t}$
      $i_c = i_*$
    end if
    // get the shares of m assets.
    $\boldsymbol{b}$ = switchEnv.state$[m+1 : 2*m]$
    $j_* = \arg\max_j \boldsymbol{a}[i_*]_j$ //select the best asset j_*.
    // action sparsification
    $a'_j = -b_j,\ \forall j \neq j_* \wedge b_j > 0$
    $a'_{j_*} = \frac{1}{p_{j_*}} \left(\omega_{cash} + \sum_{j \neq j_*} p_j b_j\right)$
  end if
  switchEnv.step$(\boldsymbol{a}')$
end for
Output: switchEnv.asset_memory[n]

sponding datasets.

5.2 DIVERSIFICATION OF AGENT POOL

To assess the diversity in trading behavior within the agent pool, we conducted experiments on six distinct test datasets, as illustrated in Figure 2. Notably, our observations revealed that no single agent consistently outperformed across all datasets. For instance, while the agent M_{mean} delivered remarkable results in the DOW and CRYPTO datasets, its performance was comparatively weaker in the FTSE dataset. However, the ensemble algorithm BDM achieved the best performance across all datasets. This experiment effectively demonstrates the behavior-diversity present within the agent pool and highlights the effectiveness of the ensemble approach in achieving superior results.

5.3 SPARSIFICATION OF ACTIONS

We present the results of using the l_0-norm $\|.\|_0$ to measure the sparsity of holding shares vectors $\boldsymbol{b}$ in Table 3. The l_0-norm $\|\boldsymbol{x}\|_0$ of a vector $\boldsymbol{x}$ is a suitable measure for the sparsity of $\boldsymbol{x}$ as $\|\boldsymbol{x}\|_0$ denotes the number of non-zero components in $\boldsymbol{x}$. For instance, on the DOW dataset, BDM's shares vectors $\boldsymbol{b}$ with $\|\boldsymbol{b}\|_0 = 1$ in 497 out of 504 trading periods.

Table 3: The l_0-norm of holding shares vectors $\boldsymbol{b}$.

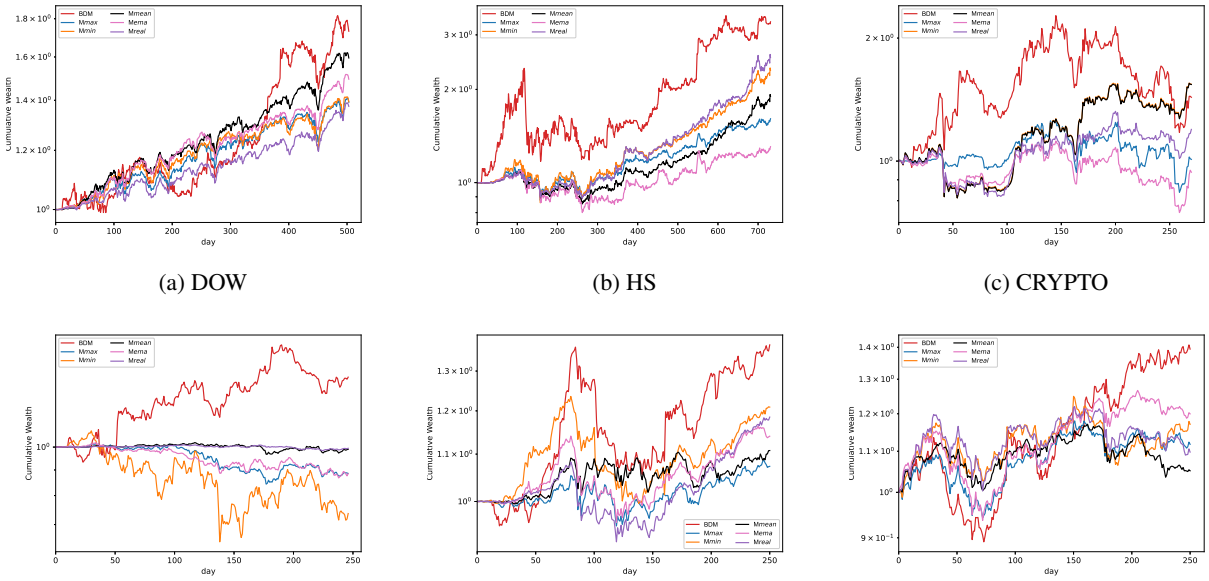
l_0-norm		DOW	HS	CRYPTO	HK	NYSE	FTSE
M_{max}	$\|\boldsymbol{b}\|_0 = 1$	0	0	0	0	0	0
	$\|\boldsymbol{b}\|_0 > 1$	504	731	271	247	251	252
M_{min}	$\|\boldsymbol{b}\|_0 = 1$	0	0	0	0	0	0
	$\|\boldsymbol{b}\|_0 > 1$	504	731	271	247	251	252
M_{mean}	$\|\boldsymbol{b}\|_0 = 1$	0	0	0	0	0	0
	$\|\boldsymbol{b}\|_0 > 1$	504	731	271	247	251	252
M_{ema}	$\|\boldsymbol{b}\|_0 = 1$	0	0	0	0	0	0
	$\|\boldsymbol{b}\|_0 > 1$	504	731	271	247	251	252
M_{real}	$\|\boldsymbol{b}\|_0 = 1$	0	0	0	0	0	0
	$\|\boldsymbol{b}\|_0 > 1$	504	731	271	247	251	252
BDM	$\|\boldsymbol{b}\|_0 = 1$	497	724	264	240	244	245
	$\|\boldsymbol{b}\|_0 > 1$	7	7	7	7	7	7

Table 3 provides confirmation of the effectiveness of Eq. (7) in generating sparse solution vectors for the BDM algorithm. In Section4.4, non-trading periods are determined based on the past cumulative wealth performance over a period of $D = \{2, 5, 7\}$ days. Consequently, agent selection and sparse operations commence only after the initial $\max(D) = 7$ days of trading, utilizing solely the M_{real} agent, as indicated by the last row of Table 3.

5.4 PERFORMANCE COMPARISON

This section presents a comprehensive performance comparison between the proposed BDM algorithm and eleven other algorithms. The parameter settings for training BDM agents are as follows: the update step size of 2048, the entropy coefficient of $c_e = 0.01$, the learning rate of 0.0003, and the batch size of 128. In terms of inference on BDM agents, the parameter settings include the decay factor of $\beta = \frac{2}{7}$ for the M_{ema} type of agents, the agent switching coefficient of $\alpha = 0.5$, and the candidate set of non-trading periods $D = \{2, 5, 7\}$.

The eleven algorithms are as follows: the uniform buy-and-hold (BAH) [Li and Hoi, 2014], the trend peak price tracing



(a) DOW (b) HS (c) CRYPTO

(d) HK (e) NYSE (f) FTSE

Figure 2: Cumulative wealth dynamics of agent pool S.

(TPPT) [Dai et al., 2022], the integrate expert strategies based on mean reversion and trading volume (MRvol) [Lin et al., 2024], the short-term sparse portfolio optimization base on l_o-norm (SSPO_l_0) [Wang et al., 2023], an online expert aggregation (OEA) [He and Yang, 2022], the deep reinforcement learning model combining long short-term memory (LSTM) deep neural network and A2C reinforcement learning algorithm (LSTM-A2C) [Zhang et al., 2022a], the self-attention based deep direct recurrent reinforcement learning with hybrid loss (SA-DDR-HL) [Kwak et al., 2023], a twin delayed deep deterministic policy gradient (TD3) [Kabbani and Duman, 2022], the imitative recurrent deterministic policy gradient (iRDPG) [Liu et al., 2020], a highly efficient gradient boosting decision tree (lightGBM) [Ke et al., 2017], and a double ensemble algorithm (DE) [Zhang et al., 2020a]. The parameter settings for these algorithms are specified according to the recommendations provided by their respective authors.

In Table 4, the performance metrics for all algorithms on six test datasets are presented, with the transaction cost set at 0.1%. The subsequent sections analyze the results using performance indices.

5.4.1 Cumulative Wealth

Cumulative wealth (CW) serves as a pivotal metric for assessing the long-term performance of an algorithm. It measures the ratio of the final cumulative wealth ω_n to the initial wealth ω_0 by calculating the product of $\mathbf{x}_t^\intercal \mathbf{b}_t$ across the entire time horizon n:

$$\text{CW} := \frac{\omega_n}{\omega_0} = \prod_{t=1}^{n} \mathbf{x_t}^\intercal \mathbf{b_t}. \tag{9}$$

Here, the relative price $\mathbf{x_t}$ is defined as $\mathbf{x_t} = \frac{\mathbf{p_t}}{\mathbf{p_{t-1}}}$. Table 4 demonstrates the CW values for various algorithms, providing compelling evidence of the outstanding performance of the BDM algorithm. It consistently ranks first on five out of the six datasets based on this metric. Notably, on the HS dataset, BDM achieves an impressive CW of 3.31, which is more than two times higher than the second-place algorithm iRDPG's CW of 1.59. The exceptional performance of BDM can be attributed to its diverse agent pool and the ability to generate sparse actions, enabling effective adaptation to uncertain trading environments.

5.4.2 Sharpe Ratio

Sharpe ratio (SR) assesses the performance of a portfolio relative to risk-free assets after adjusting for risk. It is defined as $\text{SR} := \frac{\mathbb{E}[R-R_f]}{\sigma[R-R_f]}$, where R represents the random variable of the portfolio's daily return, and R_f denotes the daily return of a risk-free asset, which is assumed to be 0 in this paper. On the HS and HK datasets, BDM stands out with the highest SRs. It secures the second position on the CRYPTO and FTSE datasets, trailing the leading algorithm by mere 0.1 and 0.08, respectively. It is important to note that the SR metric uses the variance of daily returns $\sigma[R - R_f]$ as a measure of risk, without differentiating between positive and negative returns. This can lead to an overestimation of

Table 4: Algorithm performance comparison (The highest score in each row is highlighted in bold).

DataSets	Metrics	BAH	BDM	TPPT	MRvol	SSPO_l_0	OEA	LSTM-A2C	SA-DDR-HL	TD3	iRDPG	lightGBM	DE
DOW	CW	1.53±0.00	**1.73±0.00**	1.48±0.00	1.66±0.00	1.53±0.00	1.50±0.00	1.28±0.00	1.37±0.00	1.60±0.00	1.57±0.00	1.38±0.06	1.41±0.09
	SR	1.96±0.00	1.45±0.00	0.95±0.00	1.58±0.00	1.08±0.00	1.98±0.00	1.04±0.00	1.36±0.00	**1.99±0.00**	1.91±0.00	-0.02±0.00	-0.02±0.01
	MDD (%)	6.68±0.00	13.75±0.00	19.42±0.00	10.04±0.00	13.29±0.00	**6.21±0.00**	8.26±0.00	7.73±0.00	7.82±0.00	9.16±0.00	60.73±3.45	58.57±5.15
	CAR	3.55±0.00	2.30±0.00	1.11±0.00	2.86±0.00	1.79±0.00	**3.64±0.00**	1.59±0.00	2.19±0.00	3.38±0.00	2.77±0.00	0.11±0.03	0.09±0.05
HS	CW	1.30±0.00	**3.31±0.00**	1.14±0.00	0.61±0.00	0.77±0.00	1.33±0.00	0.82±0.00	1.02±0.00	1.20±0.00	1.59±0.00	1.10±0.05	1.02±0.03
	SR	0.31±0.00	**1.14±0.00**	0.10±0.00	-0.49±0.00	-0.20±0.00	0.33±0.00	-0.12±0.00	0.15±0.00	0.37±0.00	0.70±0.00	-0.01±0.00	-0.02±0.00
	MDD (%)	47.32±0.00	49.97±0.00	45.82±0.00	58.71±0.00	50.85±0.00	42.90±0.00	48.42±0.00	**40.46±0.00**	42.37±0.00	42.88±0.00	61.16±5.24	65.79±1.56
	CAR	0.20±0.00	**1.02±0.00**	0.10±0.00	-0.26±0.00	-0.17±0.00	0.24±0.00	-0.14±0.00	0.02±0.00	0.15±0.00	0.41±0.00	0.07±0.02	0.10±0.01
CRYPTO	CW	0.92±0.00	1.43±0.00	0.28±0.00	1.14±0.00	0.43±0.00	0.90±0.00	1.35±0.00	**1.45±0.00**	0.90±0.00	0.93±0.00	0.91±0.05	0.88±0.05
	SR	-0.14±0.00	0.82±0.00	-1.69±0.00	0.28±0.00	-1.09±0.00	-0.19±0.00	0.83±0.00	**0.92±0.00**	0.09±0.00	0.21±0.00	-0.01±0.01	-0.02±0.01
	MDD (%)	35.46±0.00	48.24±0.00	77.81±0.00	**29.34±0.00**	64.75±0.00	37.58±0.00	31.08±0.00	29.66±0.00	48.54±0.00	48.17±0.00	31.14±2.07	31.51±2.21
	CAR	-0.20±0.00	0.82±0.00	-0.89±0.00	0.45±0.00	-0.84±0.00	-0.26±0.00	1.03±0.00	**1.38±0.00**	-0.19±0.00	-0.13±0.00	0.26±0.12	0.36±0.12
HK	CW	1.06±0.00	**1.37±0.00**	0.74±0.00	1.25±0.00	0.59±0.00	0.83±0.00	0.74±0.00	1.02±0.00	1.04±0.00	0.94±0.00	1.06±0.04	0.95±0.05
	SR	0.28±0.00	**1.15±0.00**	-0.44±0.00	0.69±0.00	-0.74±0.00	-0.71±0.00	-0.61±0.00	0.19±0.00	0.36±0.00	-0.38±0.00	-0.02±0.01	-0.05±0.02
	MDD (%)	17.55±0.00	20.09±0.00	48.28±0.00	21.94±0.00	52.19±0.00	31.73±0.00	44.02±0.00	**11.63±0.00**	12.89±0.00	16.66±0.00	19.77±1.19	26.26±2.57
	CAR	0.33±0.00	**1.90±0.00**	-0.54±0.00	1.14±0.00	-0.79±0.00	-0.55±0.00	-0.61±0.00	0.16±0.00	0.32±0.00	-0.36±0.00	0.33±0.19	0.61±0.16
NYSE	CW	1.15±0.00	**1.37±0.00**	1.30±0.00	1.34±0.00	1.05±0.00	1.07±0.00	1.09±0.00	1.00±0.00	1.21±0.00	1.20±0.00	1.07±0.05	1.08±0.03
	SR	1.19±0.00	1.45±0.00	0.95±0.00	1.59±0.00	0.22±0.00	0.52±0.00	0.76±0.00	0.06±0.00	**1.61±0.00**	1.27±0.00	-0.09±0.13	-0.05±0.01
	MDD (%)	11.08±0.00	22.59±0.00	16.85±0.00	13.08±0.00	16.75±0.00	11.60±0.00	15.88±0.00	13.60±0.00	**10.49±0.00**	12.90±0.00	16.43±1.4	15.68±1.24
	CAR	1.37±0.00	1.65±0.00	1.81±0.00	**2.58±0.00**	0.28±0.00	0.58±0.00	0.59±0.00	-0.01±0.00	2.04±0.00	1.53±0.00	0.54±0.15	0.59±0.11
FTSE	CW	1.16±0.00	**1.39±0.00**	0.98±0.00	0.98±0.00	1.30±0.00	1.17±0.00	1.13±0.00	1.00±0.00	1.24±0.00	1.25±0.00	1.09±0.05	1.11±0.05
	SR	0.91±0.00	1.45±0.00	-0.08±0.00	-0.10±0.00	0.99±0.00	1.03±0.00	1.03±0.00	0.10±0.00	**1.53±0.00**	1.15±0.00	-0.05±0.01	-0.04±0.02
	MDD (%)	14.83±0.00	21.48±0.00	25.10±0.00	15.98±0.00	17.42±0.00	14.30±0.00	12.59±0.00	13.46±0.00	**9.51±0.00**	16.84±0.00	39.41±3.57	37.86±3.83
	CAR	1.08±0.00	1.84±0.00	-0.09±0.00	-0.12±0.00	1.74±0.00	1.22±0.00	1.06±0.00	0.03±0.00	**2.51±0.00**	1.49±0.00	0.36±0.07	0.33±0.09

risk when there is actual growth in daily returns, resulting in a potential bias in risk-adjusted evaluation [Gregoriou and Gueyie, 2003].

5.4.3 Maximum Drawdown

Maximum Drawdown (MDD) represents the maximum observed loss from a peak to a trough of a portfolio before reaching a new peak. It is expressed $\mathrm{MDD} := \max_{t\in[1,n]} \max_{\tau\leq t} \frac{\omega_\tau - \omega_t}{\omega_\tau}$. BDM demonstrates relatively lower Maximum Drawdown (MDD) performance compared to the best-performing algorithms across the six test datasets. This can be attributed to the lower trading frequency exhibited by these algorithms, leading to a comparatively lower MDD. However, it is important to note that the Cumulative Wealth (CW) of these algorithms is significantly lower than that of BDM. For example, on the HS dataset, the SA-DDR-HL algorithm achieves the lowest MDD, which is 40.46% lower than BDM's MDD of 49.97%. However, the CW of SA-DDR-HL, at 1.02, is only around one-third of BDM's CW of 3.31. This highlights the trade-off between MDD and CW, where algorithms with lower MDDs may have comparatively lower cumulative wealth.

5.4.4 Calmar Ratio

Calmar Ratio (CAR) is a performance metric used for investment funds. It is calculated by dividing the fund's annualized return (AR) by its maximum drawdown (MDD): $\mathrm{CAR} := \mathrm{AR}/\mathrm{MDD}$. The AR is determined by the formula $\mathrm{AR} := (1+\mathrm{CR})^{\frac{252}{n}} - 1$, where $\mathrm{CR} := \frac{\omega_n}{\omega_0} - 1$ represents the cumulative return. BDM exhibits superior performance on the CAR metric compared to the MDD metric, primarily attributed to the incorporation of the profit-related metric AR in the CAR calculation formula. Specifically, BDM attains the highest CARs on HS and HK datasets, secures the second position on FTSE, and ranks third on CRYPTO. A high CAR can be achieved by efficiently minimizing MDD, signifying CAR as a measure of AR relative to the cost incurred by MDD.

6 CONCLUSION

This paper presents a novel behavior-diversified multi-agent trading algorithm (BDM) that addresses the challenges of achieving stable profits in non-stationary and "zero-sum" dynamic uncertain financial trading environments. Our approach tackles these issues from two key perspectives: behavior-diverse agent pool construction and outputting sparse investment actions. To promote behavior-diversity, we introduce a hedged structure of price-trend features, facilitating the training of agents with distinct trading behaviors. This design enables us to build an agent pool where profitable agents already exist. Additionally, we tackle sparse investment actions by devising an effective agent-selection method based on agents' long-term and short-term returns. We also incorporate an action-sparsification technique that generates accurate investment actions focusing on, at most, one asset.

Extensive experiments conducted on six diverse test datasets from financial markets affirm the superiority of BDM over state-of-the-art DRL and OPS algorithms. BDM achieves an average Compound Wealth (CW) of 1.77 and an impressive average Compound Annual Return (CAR) of 1.56, considering transaction costs exceeding 0.1%. These results exemplify the exceptional capability of BDM to generate stable profits in non-stationary and "zero-sum" dynamic uncertain financial trading environments.

References

Luca De Gennaro Aquino, Didier Sornette, and Moris S Strub. Portfolio selection with exploration of new investment assets. *European Journal of Operational Research*, 310(2):773–792, 2023.

Stephen A Berkowitz, Dennis E Logue, and Eugene A Noser Jr. The total cost of transactions on the nyse. *The Journal of Finance*, 43(1):97–112, 1988.

Min-You Chen, Chiao-Ting Chen, and Szu-Hao Huang. Knowledge distillation for portfolio management using multi-agent reinforcement learning. *Advanced Engineering Informatics*, 57:102096, 2023.

Yi Chen and Aimin Zhou. Multiobjective portfolio optimization via pareto front evolution. *Complex & Intelligent Systems*, 8(5):4301–4317, 2022.

Thomas M Cover. Universal portfolios. *Mathematical finance*, 1(1):1–29, 1991.

Hong-Liang Dai, Chu-Xin Liang, Hong-Ming Dai, Cui-Yin Huang, and Rana Muhammad Adnan. An online portfolio strategy based on trend promote price tracing ensemble learning algorithm. *Knowledge-Based Systems*, 239:107957, 2022.

Ze Yang Ding, Junn Yong Loo, Vishnu Monn Baskaran, Surya Girinatha Nurzaman, and Chee Pin Tan. Predictive uncertainty estimation using deep learning for soft robot multimodal sensing. *IEEE Robotics and Automation Letters*, 6(2):951–957, 2021.

Jacopo Fior and Luca Cagliero. A risk-aware approach to stock portfolio allocation based on deep q-networks. In *2022 IEEE 16th International Conference on Application of Information and Communication Technologies (AICT)*, pages 1–5. IEEE, 2022.

Greg N Gregoriou and Jean-Pierre Gueyie. Risk-adjusted performance of funds of hedge funds using a modified sharpe ratio. *The Journal of wealth management*, 6(3): 77–83, 2003.

Sini Guo and Wai-Ki Ching. High-order markov-switching portfolio selection with capital gain tax. *Expert Systems with Applications*, 165:113915, 2021.

Sini Guo, Jia-Wen Gu, Christopher H Fok, and Wai-Ki Ching. Online portfolio selection with state-dependent price estimators and transaction costs. *European Journal of Operational Research*, 311(1):333–353, 2023.

Botao Hao, Tor Lattimore, Csaba Szepesvári, and Mengdi Wang. Online sparse reinforcement learning. In *International Conference on Artificial Intelligence and Statistics*, pages 316–324. PMLR, 2021.

Jin'an He and Xingyu Yang. Universal portfolio selection strategy by aggregating online expert advice. *Optimization and Engineering*, pages 1–25, 2022.

Chloe Ching-Yun Hsu, Celestine Mendler-Dünner, and Moritz Hardt. Revisiting design choices in proximal policy optimization. *arXiv preprint arXiv:2009.10897*, 2020.

Dingjiang Huang, Yan Zhu, Bin Li, Shuigeng Zhou, and Steven CH Hoi. Semi-universal portfolios with transaction costs. 2015.

Yuling Huang, Chujin Zhou, Kai Cui, and Xiaoping Lu. A multi-agent reinforcement learning framework for optimizing financial trading strategies based on timesnet. *Expert Systems with Applications*, 237:121502, 2024.

Zhenhan Huang and Fumihide Tanaka. Mspm: A modularized and scalable multi-agent reinforcement learning-based system for financial portfolio management. *Plos one*, 17(2):e0263689, 2022.

Gang Ji, Jingmin Yu, Kai Hu, Jie Xie, and Xunsheng Ji. An adaptive feature selection schema using improved technical indicators for predicting stock price movements. *Expert Systems with Applications*, 200:116941, 2022.

Taylan Kabbani and Ekrem Duman. Deep reinforcement learning approach for trading automation in the stock market. *IEEE Access*, 10:93564–93574, 2022.

Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems*, 30, 2017.

Dongkyu Kwak, Sungyoon Choi, and Woojin Chang. Self-attention based deep direct recurrent reinforcement learning with hybrid loss for trading signal generation. *Information Sciences*, 623:592–606, 2023.

Zhao-Rong Lai, Dao-Qing Dai, Chuan-Xian Ren, and Ke-Kun Huang. A peak price tracking-based learning system for portfolio selection. *IEEE Transactions on Neural Networks and Learning Systems*, 29(7):2823–2832, 2017.

Zhao-Rong Lai, Dao-Qing Dai, Chuan-Xian Ren, and Ke-Kun Huang. Radial basis functions with adaptive input and composite trend representation for portfolio selection. *IEEE Transactions on Neural Networks and Learning Systems*, 29(12):6214–6226, 2018.

Zhao-Rong Lai, Liming Tan, Xiaotian Wu, and Liangda Fang. Loss control with rank-one covariance estimate for short-term portfolio optimization. *The Journal of Machine Learning Research*, 21(1):3815–3851, 2020.

Jinho Lee, Raehyun Kim, Seok-Won Yi, and Jaewoo Kang. Maps: multi-agent reinforcement learning-based portfolio management system. In *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence*, pages 4520–4526, 2021.

Kyungjae Lee, Sungjoon Choi, and Songhwai Oh. Sparse markov decision processes with causal sparse tsallis entropy regularization for reinforcement learning. *IEEE Robotics and Automation Letters*, 3(3):1466–1473, 2018.

Bin Li and Steven CH Hoi. Online portfolio selection: A survey. *ACM Computing Surveys (CSUR)*, 46(3):1–36, 2014.

Bin Li, Steven CH Hoi, Doyen Sahoo, and Zhi-Yong Liu. Moving average reversion strategy for on-line portfolio selection. *Artificial Intelligence*, 222:104–123, 2015.

Tengfei Li, Kani Chen, Yang Feng, and Zhiliang Ying. Binary switch portfolio. *Quantitative Finance*, 17(5):763–780, 2017.

Xiao Peng Li, Zhang-Lei Shi, Chi-Sing Leung, and Hing Cheung So. Sparse index tracking with k-sparsity or ε-deviation constraint via l0-norm minimization. *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

Hong Lin, Yong Zhang, and Xingyu Yang. Online portfolio selection of integrating expert strategies based on mean reversion and trading volume. *Expert Systems with Applications*, 238:121472, 2024.

Siyu Lin and Peter A Beling. An end-to-end optimal trade execution framework based on proximal policy optimization. In *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence*, pages 4548–4554, 2021.

Xiao-Yang Liu, Hongyang Yang, Jiechao Gao, and Christina Dan Wang. Finrl: Deep reinforcement learning framework to automate trading in quantitative finance. In *Proceedings of the second ACM international conference on AI in finance*, pages 1–9, 2021.

Yang Liu, Qi Liu, Hongke Zhao, Zhen Pan, and Chuanren Liu. Adaptive quantitative trading: An imitative deep reinforcement learning approach. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 2128–2135, 2020.

Hui Niu, Siyuan Li, and Jian Li. Metatrader: An reinforcement learning approach integrating diverse policies for portfolio optimization. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pages 1573–1583, 2022.

Dan Ofer, Nadav Brandes, and Michal Linial. The language of proteins: Nlp, machine learning & protein sequences. *Computational and Structural Biotechnology Journal*, 19: 1750–1758, 2021.

Antonio Riva, Lorenzo Bisi, Pierre Liotet, Luca Sabbioni, Edoardo Vittori, Marco Pinciroli, Michele Trapletti, and Marcello Restelli. Addressing non-stationarity in fx trading with online model selection of offline rl experts. In *Proceedings of the Third ACM International Conference on AI in Finance*, pages 394–402, 2022.

Russell Roberts and Shai Davidai. The psychology of asymmetric zero-sum beliefs. *Journal of Personality and Social Psychology*, 123(3):559, 2022.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Ali Shavandi and Majid Khedmati. A multi-agent deep reinforcement learning framework for algorithmic trading in financial markets. *Expert Systems with Applications*, 208:118124, 2022.

Yauheniya Shynkevich, T Martin McGinnity, Sonya A Coleman, Ammar Belatreche, and Yuhua Li. Forecasting price movements using technical indicators: Investigating the impact of varying input window length. *Neurocomputing*, 264:71–88, 2017.

Hao Wang, Wanying Zhang, Yuxin He, and Wenming Cao. l0-norm based short-term sparse portfolio optimization algorithm based on alternating direction method of multipliers. *Signal Processing*, 208:108957, 2023.

Zhicheng Wang, Biwei Huang, Shikui Tu, Kun Zhang, and Lei Xu. Deeptrader: a deep reinforcement learning approach for risk-return balanced portfolio management with market conditions embedding. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 643–650, 2021.

Hongyang Yang, Xiao-Yang Liu, Shan Zhong, and Anwar Walid. Deep reinforcement learning for automated stock trading: An ensemble strategy. In *Proceedings of the first ACM international conference on AI in finance*, pages 1–8, 2020a.

Xingyu Yang, Jin'an He, Hong Lin, and Yong Zhang. Boosting exponential gradient strategy for online portfolio selection: an aggregating experts' advice method. *Computational Economics*, 55(1):231–251, 2020b.

Xingyu Yang, Jin'an He, and Yong Zhang. Aggregating exponential gradient expert advice for online portfolio selection. *Journal of the Operational Research Society*, 73(3):587–597, 2022.

Yunan Ye, Hengzhi Pei, Boxin Wang, Pin-Yu Chen, Yada Zhu, Ju Xiao, and Bo Li. Reinforcement-learning based portfolio management with augmented asset movement prediction states. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 1112–1119, 2020.

Jianfei Yin, Ruili Wang, Yeqing Guo, Yizhe Bai, Shunda Ju, Weili Liu, and Joshua Zhexue Huang. Wealth flow model: Online portfolio selection based on learning wealth flow matrices. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, (2):1–27, 2021.

Chuheng Zhang, Yuanqi Li, Xi Chen, Yifei Jin, Pingzhong Tang, and Jian Li. Doubleensemble: A new ensemble method based on sample reweighting and feature selection for financial data analysis. In *2020 IEEE International Conference on Data Mining (ICDM)*, pages 781–790. IEEE, 2020a.

Huanming Zhang, Zhengyong Jiang, and Jionglong Su. A deep deterministic policy gradient-based strategy for stocks portfolio management. In *2021 IEEE 6th International Conference on Big Data Analytics (ICBDA)*, pages 230–238. IEEE, 2021.

Nan Zhang, Jingyuan Wang, and Mingzhong Xiao. Deep reinforcement learning trading strategy based on lstm-a2c model. In *International Conference on Advanced Algorithms and Neural Networks (AANN 2022)*, volume 12285, pages 281–287. SPIE, 2022a.

Yifan Zhang, Peilin Zhao, Qingyao Wu, Bin Li, Junzhou Huang, and Mingkui Tan. Cost-sensitive portfolio selection via deep reinforcement learning. *IEEE Transactions on Knowledge and Data Engineering*, 34(1):236–248, 2020b.

Yong Zhang, Hong Lin, Lina Zheng, and Xingyu Yang. Adaptive online portfolio strategy based on exponential gradient updates. *Journal of Combinatorial Optimization*, pages 1–25, 2022b.

BDM: Exploring Certain Profitable Opportunities in Uncertain Financial Trading Environments (Supplementary Material)

A PARAMETER SETTING OF BDM

Table 5: Overview of the validation datasets

Dataset	Region	Validate Periods	# Assets
DOW	US	02/07/2009 - 01/07/2010	29
HS	CN	05/01/2010 - 04/01/2011	29
CRYPTO	Global	07/10/2019 - 06/10/2020	29
HK	CN	27/11/2018 - 26/11/2019	29
NYSE	US	06/02/2012 - 01/02/2013	29
FTSE	UK	05/02/2018 - 01/02/2019	29

The BDM algorithm utilizes a predefined set of parameters, which are listed in Table 6.

Table 6: Parameter setting

Parameter	Value	Explain
β	$\frac{2}{7}$	Decay factor for EMA used in training agents
α	0.5	Agent switching coefficient for evaluating agents performance across long and short-term durations
D	$\{2, 5, 7\}$	Candidate set of non-trading periods

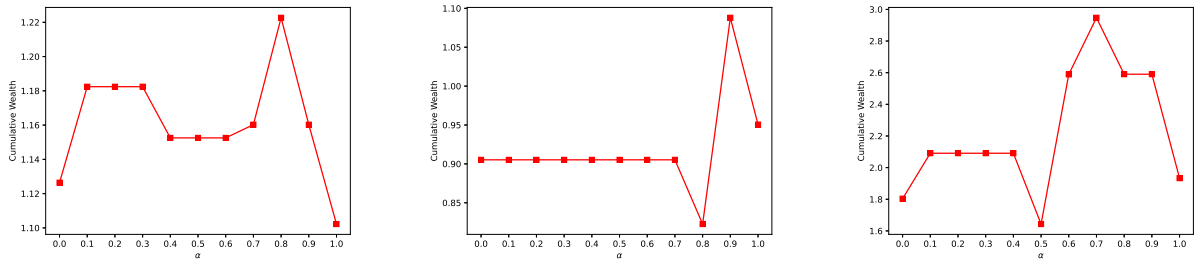
Through an evaluation of the BDM algorithm with fixed non-trading periods $d = 5$ on six validation datasets as shown in Table 5, we collect CWs for various settings of α, as depicted in Figure 3. Taking into account the algorithm's performance across all datasets, we have chosen $\alpha = 0.5$ as the default value for subsequent evaluations.

In order to determine the members of the candidate set of non-trading periods D, we evaluate the performance of the BDM algorithm fixed agent switching coefficient $\alpha = 0.5$ on six test datasets. The results, shown in Figure 4, guided our selection of $D = \{2, 5, 7\}$. This decision is based on the consideration that the inclusion of additional members in D would lead to longer run times for the BDM algorithm.

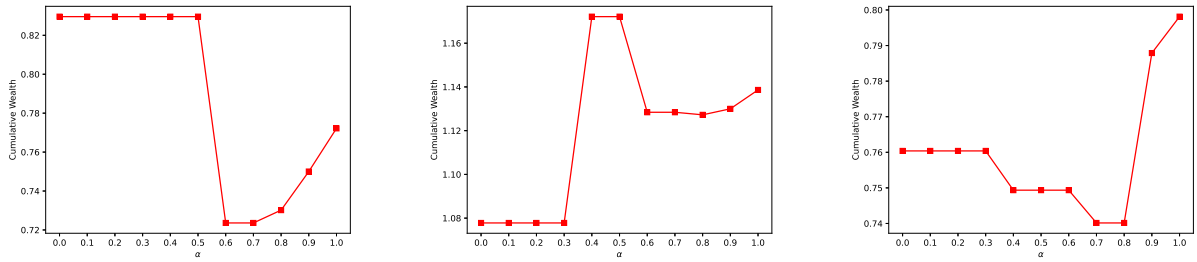
B CUMULATIVE WEALTH DYNAMICS

The dynamics of CWs for all algorithms across all periods are illustrated in Figure 5. The dynamic performance of the BDM algorithm throughout the entire transaction period can be observed from it, which cannot be gleaned from the various indicators in Table 2 alone.

On the DOW and FTSE datasets, the BDM algorithm performs exceptionally well during the middle and late stages of trading. This can be attributed to its diverse agent pool and effective trend-switching technique. Notably, the BDM algorithm demonstrates excellent overall performance on the HS dataset, with its CW exceeding that of the second-ranked algorithm by a wide margin throughout the whole trading processes.

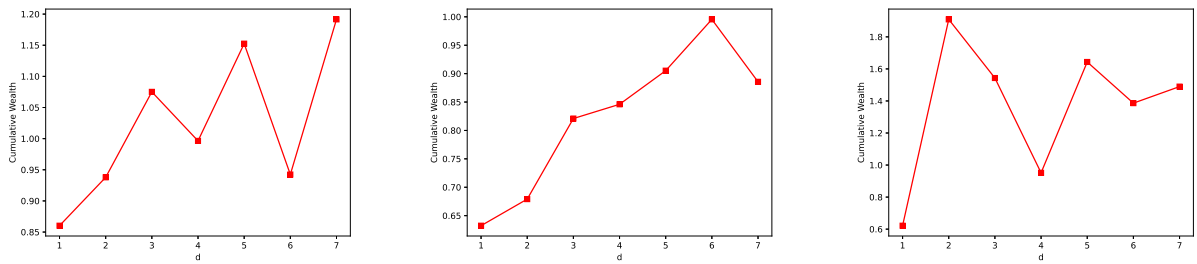


(a) DOW (b) HS (c) CRYPTO

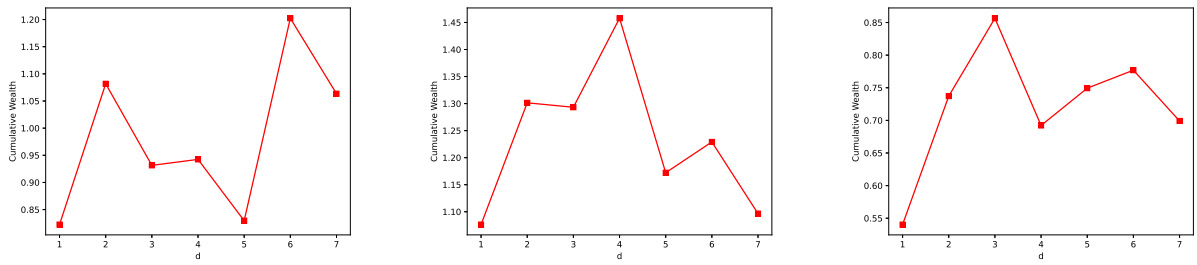


(d) HK (e) NYSE (f) FTSE

Figure 3: Exploration of agent switching coefficient α with fixed non-trading periods $d = 5$.

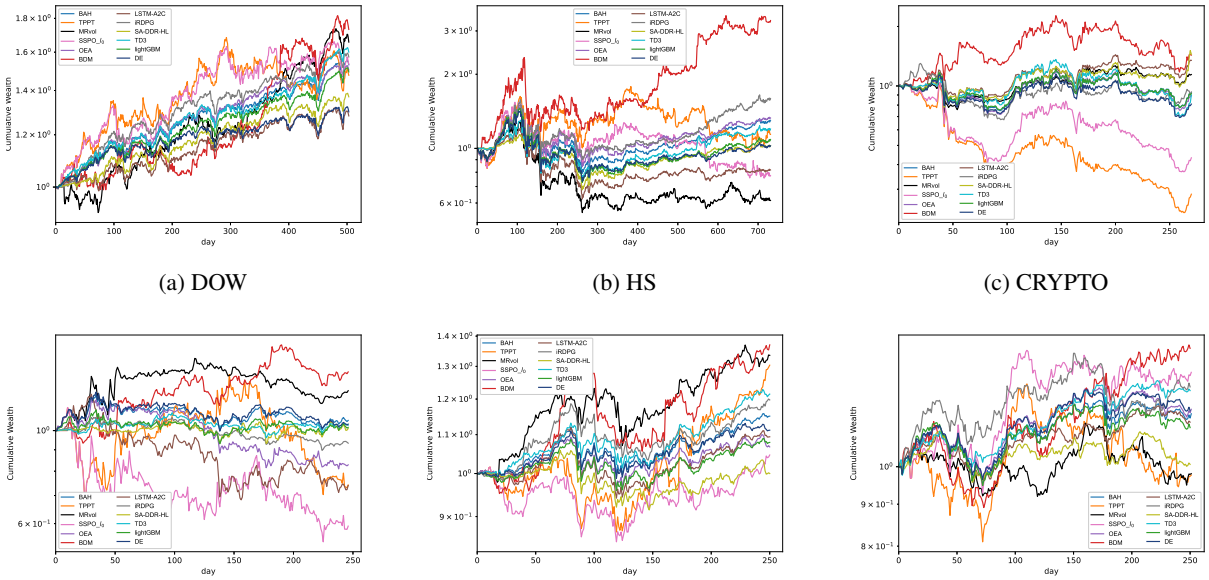


(a) DOW (b) HS (c) CRYPTO



(d) HK (e) NYSE (f) FTSE

Figure 4: Exploration of non-trading periods d with fixed agent switching coefficient $\alpha = 0.5$.



(a) DOW (b) HS (c) CRYPTO

(d) HK (e) NYSE (f) FTSE

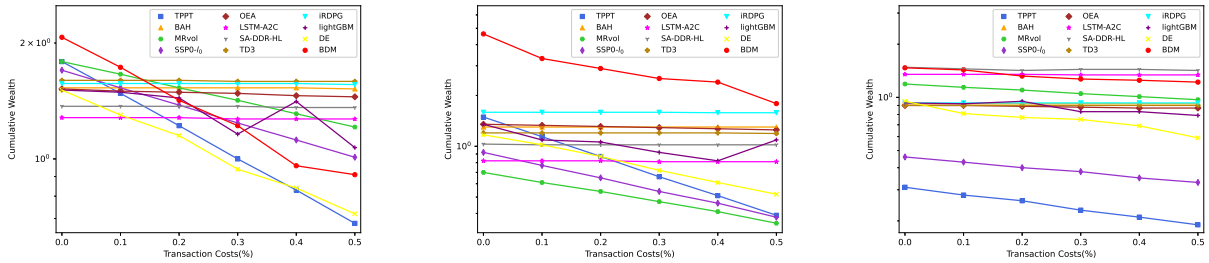
Figure 5: Cumulative wealth dynamics.

C IMPACT OF TRANSACTION COSTS

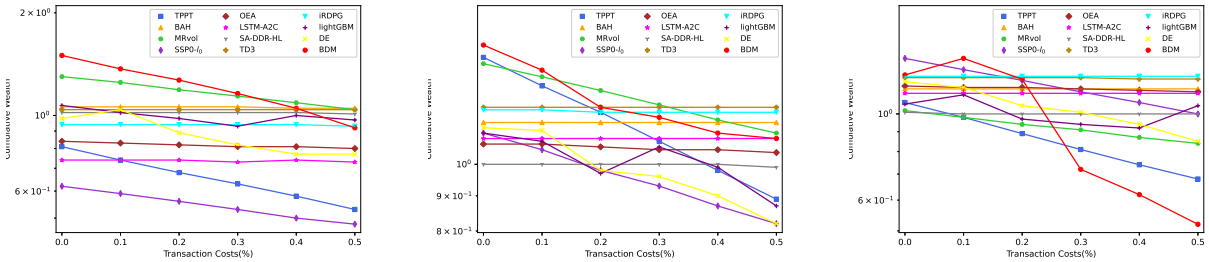
We evaluate the practical applicability of the BDM algorithm by employing the proportional transaction cost model [Lai et al., 2017, 2020] and analyzing the changes in cumulative wealth (CW) under various transaction cost scenarios. We vary transaction costs from 0.0% to 0.5%, and the results are presented in Figure 6. Notably, the BDM algorithm consistently outperforms other algorithms on the majority of test datasets, even when transaction costs reach or exceed 0.1%.

In Figure 6, the cumulative wealth dynamics of BDM reveal distinct patterns that respond to different transaction cost ratios. This phenomenon emerges from the influence of varying transaction costs on the cumulative wealth at each period, subsequently providing feedback to the agent's input state. Consequently, the agent endeavors to optimize its actions to mitigate the impact of higher transaction costs.

It is worth highlighting that empirical studies [Li et al., 2015, Berkowitz et al., 1988] have consistently shown that transaction costs in real-world financial markets are typically lower than 0.1%. However, even when transaction costs approach this threshold, the BDM algorithm consistently outperforms other algorithms in terms of cumulative wealth (CW) across the majority of datasets. This demonstrates the robustness and practical suitability of the BDM algorithm for practical transaction environments.



(a) DOW (b) HS (c) CRYPTO



(d) HK (e) NYSE (f) FTSE

Figure 6: Impact of transaction costs on cumulative wealth.