



Article

A Novel AI-Based Trading Framework for Futures Markets: Evidence from the MTX Case Study

Yu-Heng Hsieh , Chiung-Han Lai and Shyan-Ming Yuan *

Department of Computer Science, National Yang Ming Chiao Tung University, Hsinchu 30010, Taiwan; k28998989.cs11@nycu.edu.tw (Y.-H.H.); jonaslaisad.cs12@nycu.edu.tw (C.-H.L.)

* Correspondence: smyuan@nycu.edu.tw

Abstract

This study develops a novel AI-based trading framework designed to consistently generate profits across cyclical bullish and bearish futures markets. Unlike conventional strategies that rely on static rules or a single predictive model, the proposed framework introduces a dual-agent deep reinforcement learning (DRL) architecture, where one agent specializes in bullish conditions and the other in bearish conditions, while a trading decision selector dynamically predicts market regimes and allocates execution accordingly. This design enables the system to adapt to regime shifts and mitigate risks arising from market volatility and extreme events. Using Mini Taiwan Stock Exchange Index Futures (MTX) as a case study, a four-year historical backtest is conducted covering multiple disruptive periods, including the tax adjustment and the Russia–Ukraine conflict. The empirical results show that, under a monthly capital reset and loss-compensation rule with a fixed investment of TWD 500,000 per month, the proposed framework achieves an average cumulative return of 2240%, an annualized return of 109%, and a Sharpe ratio of 0.31, with the cumulative ROI exceeding twice the MTX index growth over the same period. Although the Sharpe ratio remains moderate, this outcome reflects the framework’s emphasis on directional trading and absolute return maximization, where profitable trades outweigh intermittent losses despite higher short-term volatility. These findings suggest that adaptive, regime-aware DRL architectures are particularly effective for futures trading in markets characterized by frequent trend reversals, offering both methodological innovation and practical applicability under realistic market conditions, with strong returns achieved at a moderate risk-adjusted level.

Keywords: deep learning; deep reinforcement learning; mini Taiwan stock index futures; automated trading; futures market



Academic Editor: Babacar Sène

Received: 5 January 2026

Revised: 25 February 2026

Accepted: 27 February 2026

Published: 4 March 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

Investment is a form of asset management aimed at increasing asset value over time. Among various financial instruments, futures are regarded as one of the fastest-growing and potentially most profitable tools. Due to the leverage effect, futures trading allows investors to achieve substantial returns with relatively small capital. However, consistently generating profits in the futures market is challenging, as price movements are highly unpredictable and influenced by external forces such as geopolitical events, natural disasters, and wars. These factors often trigger significant price fluctuations within short periods, amplifying both investment risks and operational complexity. Moreover, unlike stock trading, futures trading involves specific structural characteristics—such as margin

requirements, standardized contract specifications, and defined intraday trading sessions—that directly affect the feasible action space, risk accumulation, and capital management. In this study, these market features are explicitly incorporated into the environment design: the state space, allowed trading actions, and reward computation all reflect realistic margin dynamics, intraday session constraints, and position limits, ensuring that agents learn under operationally valid conditions.

To mitigate these challenges, researchers and practitioners have increasingly turned to advanced technologies, particularly artificial intelligence (AI), to design adaptive and resilient trading systems. With rapid progress in AI, numerous studies have explored its applications in finance. Deep learning models have been applied to predict future prices, while reinforcement learning (RL) has been employed to train agents capable of interacting autonomously with financial markets and making trading decisions based on learned strategies. Within deep learning, most prior research has focused on time-series models for price prediction (Al-Khasawneh et al., 2025; Gil et al., 2024; Zhao & Yang, 2023). More recently, the rise of large language models (LLMs) has spurred interest in combining LLMs with sentiment analysis to forecast market trends and price movements (Kirtac & Germano, 2024; Padmanayana & Bhavya, 2021; Y. Xu & Cohen, 2018). Nevertheless, even with accurate forecasts, profitability remains uncertain in practice, as effective strategies also require precise decisions on when to buy and sell. Compared to traditional methods, reinforcement learning offers a key advantage: agents can learn through continuous interaction with the environment, generating trading signals that adapt in real time. This capability supports the development of comprehensive trading strategies and has attracted growing attention in the financial domain, leading to a wide range of RL-based automated trading systems (Kabbani & Duman, 2022; Yang et al., 2020). Recent studies have also emphasized the importance of risk-aware RL approaches (Lillicrap et al., 2015; Sutton & Barto, 1998) and futures-specific environments (Mnih et al., 2015; Schulman et al., 2017), demonstrating that incorporating market structure, volatility constraints, and margin requirements is essential for robust performance. For example, (Qin et al., 2025) designed a risk-constrained RL agent for leveraged commodity futures, while (Z. Zhang et al., 2019) developed a deep RL framework that explicitly models intraday margin requirements and contract expirations, showing improved performance over naive agents. These works illustrate the benefit of tailoring RL strategies to the structural characteristics of futures markets.

This study focuses on the Mini Taiwan Stock Exchange Index Futures (MTX), a financial derivative reflecting the overall performance of Taiwan's stock market. MTX contracts are categorized by trading month and are available for the current month, the following two consecutive months, and the quarterly months of March, June, September, and December. Each contract expires on the third Wednesday of its designated month. In this study, we exclusively trade the current-month contract, which generally provides higher liquidity and trading volume.

The research objective is to design a robust trading framework capable of achieving stable profits across varying market conditions, including both bullish and bearish trends. To this end, we propose a novel architecture comprising two specialized trading agents and a trader selector. One agent is optimized for bullish markets, the other for bearish markets, while the trader selector—implemented using an Extended Long Short-Term Memory (xLSTM) model (Beck et al., 2024)—forecasts upcoming trends and dynamically activates the most suitable agent for execution. The framework is evaluated extensively on the MTX platform using historical market data, with the environment explicitly reflecting futures-specific constraints such as margin requirements, contract specifications, and intraday trading schedules. This ensures that agent behavior is realistic, risk assessment is credible, and learning occurs under operationally valid conditions.

The main contributions of this study are threefold:

- **Adaptive Trading Framework:** This study proposes an adaptive trading architecture that integrates two deep reinforcement learning (DRL) agents specialized for bullish and bearish market conditions. Market regimes are implicitly characterized through predicted return–risk dynamics and volatility patterns, while an xLSTM-based trader selector dynamically allocates trading actions across regimes, enabling effective adaptation to non-stationary financial markets.
- **Practical and Deployable System:** An automated trading system is developed and deployed in the Taiwan Futures Exchange (MTX) environment, demonstrating the practical feasibility and robustness of the proposed framework under real-world trading constraints, including market volatility and transaction dynamics.
- **Return-Oriented Performance Evaluation:** Extensive backtesting is conducted across multiple market scenarios and exchange-traded funds (ETFs) using both return-based and risk-adjusted performance metrics, such as cumulative return, Sharpe ratio, and maximum drawdown. The experimental results show that the proposed framework achieves substantially higher cumulative returns compared with the MTX index and benchmark ETFs, albeit at the cost of increased return volatility, resulting in a moderate Sharpe ratio. This highlights a deliberate trade-off favoring return maximization in aggressive and speculative trading settings.

The remainder of this paper is organized as follows. Section 2 reviews related work. Section 3 introduces the proposed methodology, including system architecture and learning approach. Section 4 presents and analyzes experimental results. Section 5 concludes with a summary of key findings and discusses directions for future research.

2. Background and Related Work

In this chapter, we will first provide theoretical background related to our research in Section 2.1 and discuss researches related to futures prediction in Section 2.2.

2.1. Background

2.1.1. Reinforcement Learning (RL)

Reinforcement learning (RL), unlike supervised or unsupervised learning, represents a distinct paradigm in machine learning. In RL, an autonomous agent continuously interacts with its environment and learns through reward-contingent feedback with the objective of maximizing cumulative long-term returns (Sutton & Barto, 1998). At each step, the agent selects an action based on the current state, which leads to a new state and an immediate reward. Over repeated interactions, the agent gradually learns which actions are most effective in optimizing long-term expected outcomes across diverse scenarios.

In recent years, numerous RL algorithms have been developed to improve sample efficiency, enhance training stability, and boost overall performance in complex and dynamic environments. Representative methods include Deep Deterministic Policy Gradient (DDPG) (Lillicrap et al., 2015), Deep Q-Networks (DQN) (Mnih et al., 2015), and Proximal Policy Optimization (PPO) (Schulman et al., 2017), each demonstrating strengths in different domains.

2.1.2. Proximal Policy Optimization

Proximal Policy Optimization (PPO) (Schulman et al., 2017), introduced by John Schulman, is a reinforcement learning algorithm derived from the policy gradient framework. PPO improves upon Trust Region Policy Optimization (TRPO) (Schulman et al., 2015a) by simplifying implementation while preserving stability and robustness. Its key innovation lies in the use of a clipped surrogate objective, which restricts the magnitude of policy

updates, thereby ensuring stable learning throughout the training process. The objective function is defined as follows in Equation (1):

$$L^{CLIP}(\theta) = \hat{E}_t[\min(\text{clip}(r_t(\theta), 1 - \epsilon, 1 + \epsilon)\hat{A}_t, r_t(\theta)\hat{A}_t)] \quad (1)$$

Here, $r_t(\theta)$ denotes the probability ratio between the new and old policies under the same state–action pair, formally defined as:

$$r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{add}}(a_t|s_t)},$$

where θ represents the parameters of the policy. The advantage function $\hat{A}_t$ measures how much better or worse a given action is compared to the expected performance of the policy, and is typically estimated using Generalized Advantage Estimation (GAE) (Schulman et al., 2015b) to balance bias and variance.

The clipping mechanism constrains $r_t(\theta)$ within the range $[1 - \epsilon, 1 + \epsilon]$, effectively preventing overly large policy updates. By limiting drastic changes, PPO achieves a balance between exploration and stability, making it one of the most widely adopted and effective algorithms in modern reinforcement learning.

2.1.3. Extended Long Short-Term Memory

Extended Long Short-Term Memory (xLSTM) (Beck et al., 2024), proposed by the original authors of Long Short-Term Memory (LSTM) (Hochreiter & Schmidhuber, 1996), addresses several limitations of the traditional LSTM architecture, such as the inability to revise stored information, limited memory capacity, and lack of parallelizability. To overcome these challenges, the authors introduced two novel variants: sLSTM and mLSTM.

The sLSTM incorporates exponential gates, replacing the conventional input gate activation with an exponential function. This modification allows the model to revise previously stored information, enhancing flexibility compared to standard LSTMs. Since exponential functions can produce unbounded values, the architecture introduces normalized states to stabilize the model and prevent uncontrolled growth during training.

In contrast, the mLSTM expands memory capacity by introducing a matrix-based memory structure. It incorporates key, value, and query inputs that are independent of the previous hidden state, which not only increases capacity but also enables parallel execution. This key–value–query design is reminiscent of the attention mechanism in Transformer models (Vaswani et al., 2017), while still being embedded within a recurrent neural network framework.

By integrating these two components with additional projection layers, the authors introduced residual sLSTM blocks and residual mLSTM blocks, as illustrated in Figures 1 and 2. Stacking these residual blocks constructs the complete xLSTM architecture. This design significantly improves memory capacity, stability, and computational efficiency, establishing xLSTM as a powerful extension of the original LSTM model.

2.2. Related Work

This section reviews previous research on the application of reinforcement learning (RL) and deep learning techniques in financial markets.

2.2.1. Financial Applications of Reinforcement Learning

In recent years, a growing body of research has investigated the application of reinforcement learning (RL) paradigms to financial markets. Because RL derives trading policies through iterative interaction with the environment and is particularly well suited

to volatile, decision-critical domains—such as financial trading, where the objective is to optimize cumulative returns—it has emerged as a powerful methodology for this field.

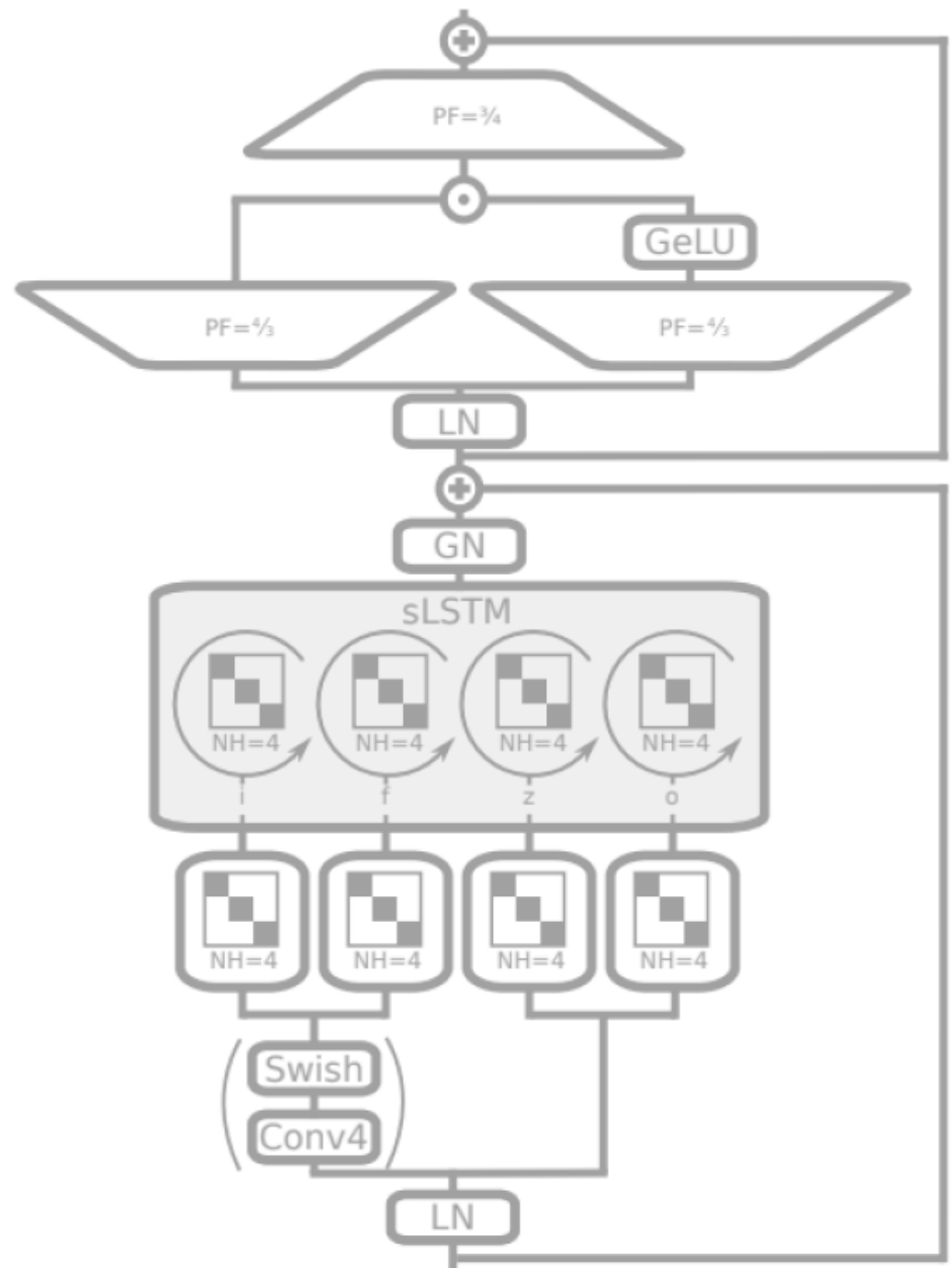


Figure 1. Residual sLSTM Block (Beck et al., 2024).

Several studies (Liu et al., 2020; M. Xu et al., 2024) adopt the Recurrent Deterministic Policy Gradient (RDPG) (Heess et al., 2015) combined with imitation learning to train agents that replicate established trading strategies, such as the Dual Thrust strategy. By incorporating imitation learning, these approaches achieve a more effective balance between exploration and exploitation, thereby improving trading performance and adaptability. Experimental results demonstrate that these methods consistently outperform traditional rule-based strategies.

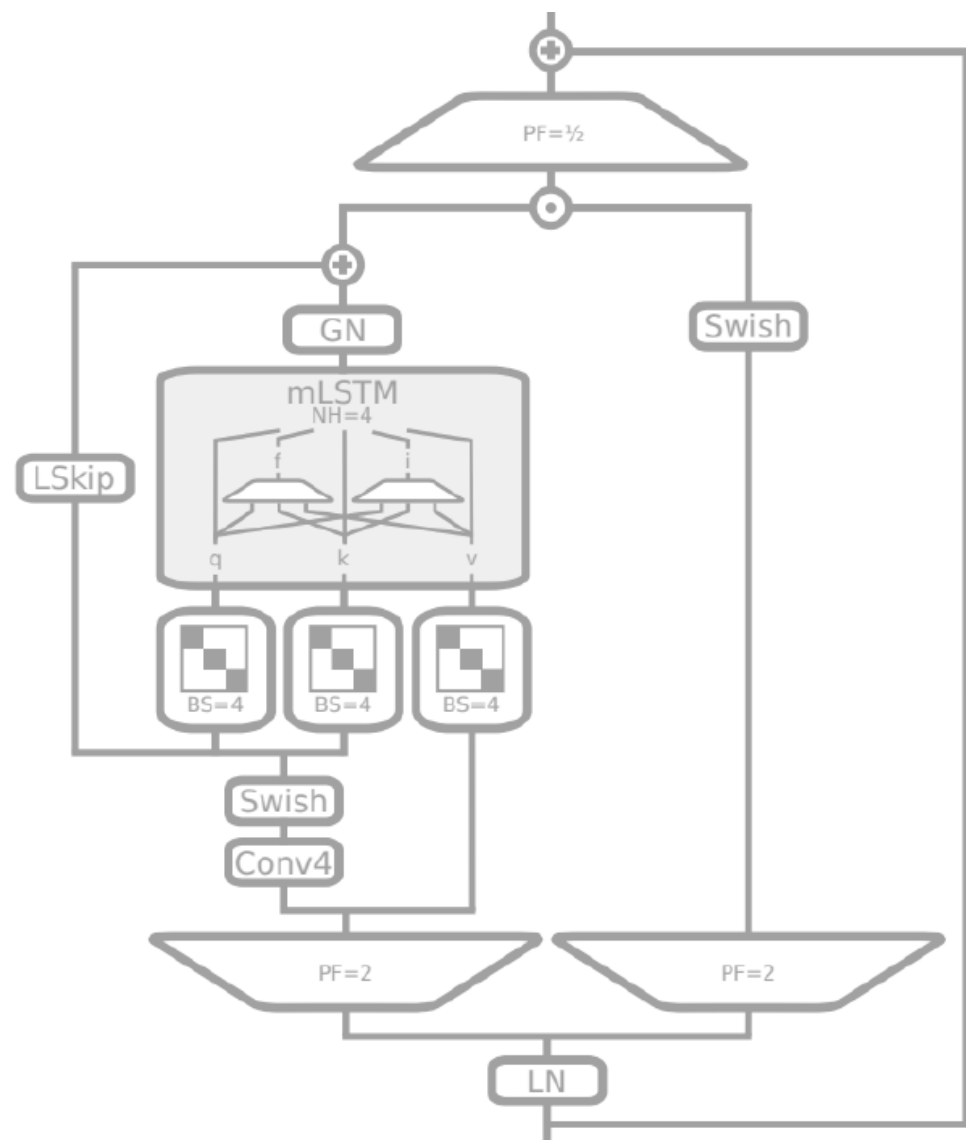


Figure 2. Residual mLSTM Block (Beck et al., 2024).

In (Heess et al., 2015), a margin trading framework was proposed to better reflect realistic market conditions by explicitly modeling margin constraints. The framework was evaluated using multiple RL algorithms, including Proximal Policy Optimization (PPO) and Deep Deterministic Policy Gradient (DDPG), with stocks from the Dow Jones Industrial Average (DJIA) serving as trading assets. The results indicate that the margin trader framework combined with PPO achieves superior performance in terms of Sharpe ratio, annualized return, and cumulative return compared with alternative methods.

To address the instability and performance limitations of individual RL algorithms, subsequent studies have explored ensemble-based approaches. In (Yang et al., 2020), multiple agents trained using PPO, Advantage Actor–Critic (A2C), and DDPG were evaluated on a quarterly basis across 30 Dow Jones stocks. For each quarter, the agent with the highest Sharpe ratio was selected for deployment in the following period. Empirical results show that this ensemble strategy consistently outperforms both individual agents and benchmark methods in terms of Sharpe ratio and cumulative return.

Further progress is reported in (Yu et al., 2023), which proposes two novel trading strategies: Nested RL and the Weighted Random Strategy with Confidence (WRSC). Nested RL adopts a hierarchical two-layer structure in which a first-layer agent selects among

three RL models—A2C, DDPG, and Soft Actor–Critic (SAC)—and the selected model is subsequently executed by the second-layer agent. In contrast, WRSC selects the agent with the highest annualized return if it exceeds a predefined confidence threshold; otherwise, an agent is chosen at random. Empirical evaluations conducted on stock markets in the U.K., U.S., and Japan demonstrate that both strategies outperform baseline approaches.

Despite the demonstrated effectiveness of reinforcement learning in financial trading, most existing studies primarily focus on equity markets or model leveraged trading in a simplified manner. Futures trading, however, exhibits distinct structural characteristics, including margin requirements, contract specifications, leverage effects, and intraday trading sessions, all of which substantially influence risk accumulation and the set of feasible trading actions. Only a limited number of studies explicitly incorporate these features into RL environment design, resulting in a noticeable gap between academic models and practical futures trading systems.

In parallel, risk-aware reinforcement learning has gained increasing attention in financial applications. Prior work integrates risk-related objectives—such as volatility penalties, drawdown constraints, or risk-adjusted performance measures (e.g., Sharpe ratio)—into the learning process to balance profitability and stability. Nevertheless, many of these approaches rely on a single trading policy or a static agent configuration, which may struggle to adapt to frequent regime shifts between bullish and bearish market conditions, particularly in highly leveraged futures markets.

2.2.2. Financial Applications of Deep Learning

Deep learning techniques are widely employed for predicting future trends and prices in financial markets. In (Gil et al., 2024), the authors explored various time series models, including Temporal Convolutional Networks (TCN) (Lea et al., 2017) and xLSTM (Beck et al., 2024), as well as other architectures (Challu et al., 2023; Das et al., 2023; Lim et al., 2021; Oreshkin et al., 2019), to forecast market prices and trends. To enhance predictive accuracy, denoising techniques were applied to the input data, enabling the models to better capture meaningful patterns. Results indicated that xLSTM outperformed all competing models across multiple evaluation metrics.

To further improve accuracy and generalization, many studies have integrated different deep learning models to exploit their complementary strengths. For instance, (J. Zhang et al., 2023) proposed a hybrid architecture that combines convolutional neural networks (CNN) with bidirectional LSTM (BiLSTM) (Schuster & Paliwal, 1997), augmented by a self-attention mechanism that dynamically re-weights feature representations. This model, referred to as CNN-BiLSTM-Attention, consistently outperformed traditional approaches across diverse financial datasets.

Similarly, (Wang, 2023) introduced TCN into the Transformer architecture to create MTRAN-TCN, which was subsequently combined with BiLSTM to form the BiLSTM-MTRAN-TCN framework. This composite model integrates the global attention mechanism of Transformers, the bidirectional sequence modeling capability of BiLSTM, and the strong sequential dependency capture of TCN, making it particularly well-suited for stock price prediction. Experimental results across multiple financial instruments demonstrated that BiLSTM-MTRAN-TCN outperformed mainstream models in most cases, achieving both high predictive accuracy and strong generalization ability.

2.2.3. Distinctiveness of the Proposed Approach

To the best of the authors' knowledge, existing reinforcement learning-based approaches to financial trading predominantly rely on a single agent to govern all trading decisions, or adopt ensemble methods that aggregate multiple agents without explicitly

distinguishing trading roles based on position orientation. In contrast, no prior study has explicitly designed separate agents dedicated to independently managing long and short positions. Our approach fundamentally departs from previous work by assigning two specialized agents to handle each trading direction, enabling role-specific learning and decision-making tailored to distinct market stances and risk profiles.

This role-specialized design is particularly well suited to leveraged futures trading environments, where long and short positions exhibit asymmetric dynamics, margin requirements, and risk characteristics. Rather than pursuing incremental architectural refinements or direct performance comparisons with existing algorithms, our work introduces a structurally novel multi-agent framework that emphasizes a clear division of trading responsibilities. This perspective complements prior research by enhancing practical trading realism and offering a new paradigm for incorporating position-aware specialization into reinforcement learning-based trading systems.

3. Methodology

3.1. Framework Overview

The proposed framework is composed of two reinforcement learning agents—a long agent and a short agent—together with a trader selector. The long agent is specialized for bullish market conditions, seeking to exploit upward price trends, while the short agent is designed for bearish environments, focusing on downward movements. The trader selector analyzes historical daily technical indicators to identify the prevailing market trend—specifically, whether prices are expected to increase or decrease on the following day—and assigns the appropriate trading agent for each trading day. The technical indicators used are described in Section 3.3. As illustrated in Figure 3, once the agent is selected, the environment supplies it with essential input features, including 15 min closing prices, available capital, the number of held contracts, and a set of technical indicators.

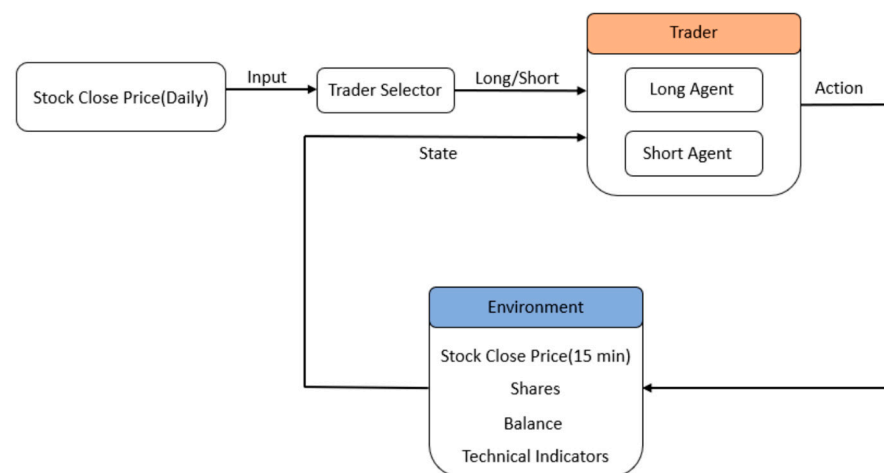


Figure 3. Proposed Framework.

3.2. RL Agent

3.2.1. Future Market Environment

The trading environment is implemented using the Python 3.9.2. Gym library (Towers et al., 2024), which provides a unified API for constructing custom environments. The design of the environment is defined as follows:

1. State Space (SS).

The state space consists of seven features: Balance, Lots, Close, Moving Average Convergence Divergence (MACD), Relative Strength Index (RSI), Commodity Channel Index (CCI), and Average Directional Index (ADX).

- Balance: The amount of capital available for investment at time t .
- Lots: The number of contracts held at time t .
- Close: The futures closing price.
- MACD: Measures momentum and trend direction, useful for detecting trend reversals.
- RSI: Identifies potential reversal points and overbought/oversold conditions.
- CCI: Captures deviations of price from its average, providing buy/sell signals.
- ADX: Evaluates the strength of a trend.

Except for Balance and Lots, all technical indicators are represented using a historical window of 76 time steps. This window was chosen based on empirical experiments balancing predictive performance and computational efficiency, and corresponds approximately to one full trading day of intraday data. Each indicator thus contributes 76 values to the state vector, resulting in a total of 328 dimensions. Importantly, all values are computed strictly from information available up to the current time t , ensuring that no future information is used and preventing look-ahead bias in the training and evaluation of trading agents.

2. Action Space (AS).

A discrete action space ranging from -100 to 100 is adopted. Positive values denote buy actions, with the magnitude specifying the number of lots purchased; negative values denote sell actions, with the magnitude specifying the number of lots sold; and 0 indicates a hold action.

3. Action Masking.

Action masking (Huang & Ontañón, 2020) is applied to exclude invalid actions from the agent's choices. For instance, if the available balance allows purchasing only one lot, actions that exceed this limit are masked. This technique enhances learning efficiency by restricting the action set to feasible operations.

4. Reward Function (RF).

The reward function is based on the profit or loss from each trade. Losses are penalized with a factor of 1.5 to discourage poor decisions. Two reward functions are used:

Long-position agent (Equation (2)):

$$reward = \begin{cases} (Close_t - bid\ price), & \text{if } (Close_t - bid\ price) \geq 0 \\ (Close_t - bid\ price) \times 1.5, & \text{otherwise} \end{cases} \quad (2)$$

Short-position agent (Equation (3)):

$$reward = \begin{cases} (bid\ price - Close_t), & \text{if } (bid\ price - Close_t) \geq 0 \\ (bid\ price - Close_t) \times 1.5, & \text{otherwise} \end{cases} \quad (3)$$

The bid price list records entry prices when opening long or covering short positions. Rewards are issued only when trades are closed. For example, if the long agent buys two lots at time t , the bid list stores two closing prices. When one lot is sold at $t + 1$, the first entry is retrieved to compute the reward.

5. Terminal Condition.

When the contract expiration date is reached, all open positions are liquidated automatically, and the environment resets the balance and lots before switching to the next month's contract.

6. Margin and Transaction Fees.

To improve realism, margin requirements and transaction fees are incorporated. The transaction fee for MTX contracts is defined as:

$$\text{Transaction Fee} = \text{Contract Value} \times 0.002\% \quad (4)$$

Margin rules include:

- Initial Margin (IM): Minimum funds required to open a new position.
- Maintenance Margin (MM): Minimum balance to sustain an open position; if the account falls below this threshold, positions are liquidated.
- Settlement Margin (SM): A fee collected by the Taiwan Futures Exchange from brokers to mitigate default risks.

For investors, only IM and MM are enforced. The agent may open a position only if its balance covers the IM, and it must liquidate positions if the balance falls below the MM to ensure compliance.

3.2.2. Datasets

1. Raw Data.

This study employs MTX futures as the trading instrument. The primary dataset was sourced from an anonymous user on Pixnet and is used solely for academic research, given its unofficial nature. To supplement missing records from September 2024 to December 2024, additional data were retrieved via the Shioaji API 1.3.2. provided by SinoPac Securities, Taiwan. The final dataset spans from 29 December 2017 to 3 June 2025. Data prior to December 2017 were excluded, as MTX night trading sessions were introduced only around that time, ensuring that the model learns from a consistent market environment covering both day and night trading hours.

The dataset was divided into training (29 December 2017–21 December 2020) and testing (22 December 2020–18 December 2024) intervals. Table 1 presents a sample of the data. This study adopts 15 min K-bar data instead of daily K-bars, since daily data provide only about 20 trading opportunities per month. During periods of high volatility, such limited frequency may hinder timely adjustments, leading to missed opportunities or larger risks.

Validation was restricted to the MTX futures market, as 15 min K-bar data from foreign markets are difficult to obtain. In contrast, such data are readily available in Taiwan through APIs offered by securities firms.

2. Synthetic Data.

An imbalance between upward and downward trends was observed in the dataset: from 2018 to 2020, 26 months exhibited upward trends while only 10 months were downward. To mitigate this bias, synthetic data were generated by reversing selected segments—swapping open and close prices and inverting their chronological order. This process effectively transforms upward into downward trends, enriching the dataset with diverse market conditions. Figure 4 illustrates this effect, where subfigure (a) shows the original October 2019 chart, and subfigure (b) presents its reversed version.

Table 1. Time Periods of Each Trader Selector.

Selector	Training	Application
2021 H1	January 2010~December 2021	January 2021~June 2021
2021 H2	July 2010~June 2021	July 2021~December 2021
2022 H1	January 2011~June 2021	January 2022~June 2022
2022 H2	July 2011~December 2021	July 2022~December 2022
2023 H1	January 2012~June 2022	January 2023~June 2023
2023 H2	July 2012~December 2022	July 2023~December 2023
2024 H1	January 2013~June 2023	January 2024~June 2024
2024 H2	July 2013~June 2024	July 2024~December 2024
2025 H1	January 2014~December 2024	January 2025~June 2025

Note: H1 and H2 denote the first half (January–June) and the second half (July–December) of each year, respectively. The training period represents the historical data used to train the trader selector model, and the application period denotes the out-of-sample testing interval.

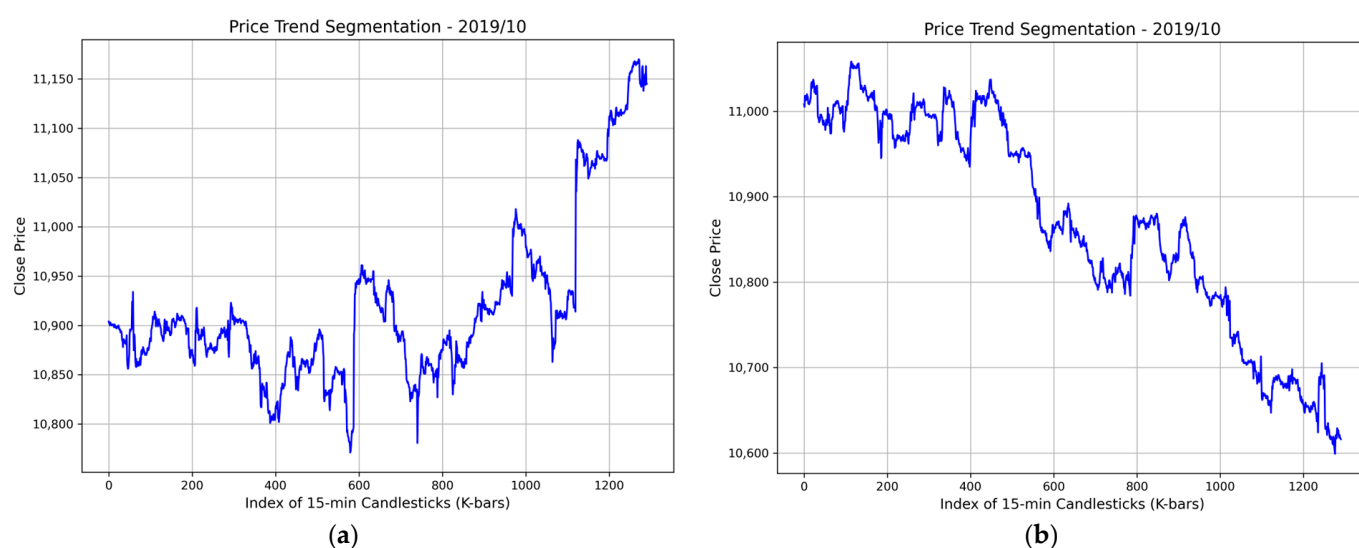


Figure 4. Visualization of original and reversed line charts for October 2019. (a) The original data, (b) The synthetic data.

3.2.3. Data Preprocessing

To ensure feature consistency, Min–Max normalization was applied, scaling all features to the range [0, 1]. This reduces the impact of differing magnitudes, stabilizes training, and accelerates convergence during optimization.

3.3. Trader Selector

The trader selector primarily employs the xLSTM model and the data preprocessing techniques proposed by Gonzalo Lopez Gil (Beck et al., 2024). Their approach applies filtering methods to remove noise from stock market data, providing the model with cleaner input signals. In this study, we adopt this methodology and apply it to MTX daily data spanning from 2010 to 2025, including open, high, low, and close prices, as well as the 20-day moving average, K, D, and MFI indicators. The trader selector operates as a classification model, outputting either a bullish or bearish trend. Based on this prediction, it selects the appropriate trading agent to execute trades on the following day. All input features are computed solely from information available up to the current day, ensuring that the model avoids look-ahead bias and makes decisions using only historically observable data.

Data Splitting and Training Procedure

We followed the (Al-Khasawneh et al., 2025; Biswas et al., 2025; Ge, 2025) method to split our datasets. The dataset is divided into three periods: the training set (January 2010–June 2020), and the application set (January 2021–June 2021). To improve model adaptability to unseen data and better simulate real-world deployment, a sliding window training strategy is adopted. After each training cycle, the training, and application sets are shifted forward by six months. This process continues until December 2024, resulting in eight independently trained Trader Selectors, each applied to a half-year period from January 2021 to December 2024. The detailed training, testing, and deployment periods for each Trader Selector are summarized in Table 1.

4. Experiments Results

4.1. Experiment Setup and Methods

4.1.1. Daily Trading Flow

Each daily K-bar, in accordance with exchange regulations, spans the night session of the preceding day through the day session of the current day. Consequently, the opening time of the night session is regarded as the start of each trading day. The overall trading procedure is as follows:

At the beginning of each night session, historical daily market data are input into the trader selector to determine which agent will be employed for the day. If the selected agent differs from the one used on the previous day, all existing positions are closed before trading begins; if the same agent is chosen, trading continues uninterrupted. Subsequently, every 15 min, historical 15 min market data are provided to the selected agent, which outputs the corresponding trading action. This cycle continues until the expiration date of the current month's contract.

The trading workflow is summarized in Figure 5 and Algorithm 1, which depict the complete process from agent selection and position management to continuous trading decisions at 15 min intervals. Upon contract expiration, all open positions are automatically closed, signaling the completion of a trading cycle. Algorithm 1 provides a step-by-step description of the proposed trading framework.

Algorithm 1: Trader Selector and Daily Trading Framework

1. Begin daily trading session (night session starts)
 2. Obtain historical daily data and perform denoising
 3. Select a trader agent for the day using the Trader Selector
 4. If the selected agent is the same as the prior day:
 - a. Retrieve historical 15 min K-line data
 - b. Selected agent decides trading action based on the data
 4. Else:
 - a. Close currently held position from the previous agent
 - b. Selected agent decides trading action
 5. Execute trading action
 6. Update account and market status
 - Check if today's early trading session has ended
 7.
 - a. If yes, loop back to step 1 for the next trading day
 - b. If no, continue monitoring until session ends
-

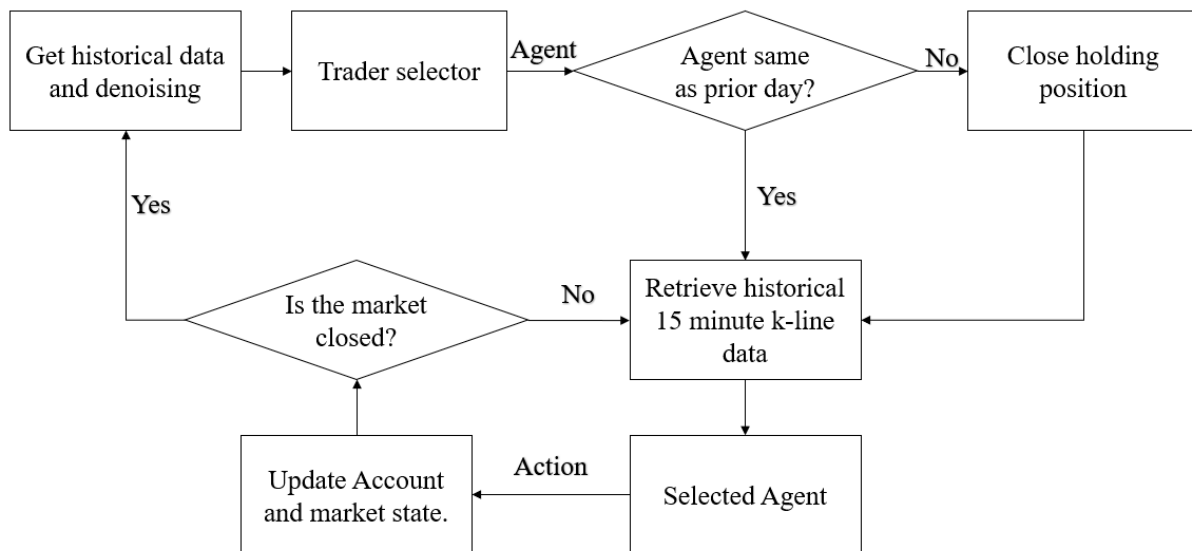


Figure 5. Workflow of the proposed trading framework.

4.1.2. Experiment Settings

In this study, all trading methods are initialized with a capital of TWD 500,000. At the beginning of each month, the available trading capital is reset to TWD 500,000, regardless of the performance in the previous month. Any losses incurred during the prior month are replenished to restore the initial capital level. The replenished amount is treated as additional capital injection and is incorporated into the return calculation to avoid overstating performance.

Moreover, all methods account for margin requirements, with margin levels dynamically adjusted according to the annual specifications announced by the Taiwan Futures Exchange.

In our simulation, we account for the possibility of execution slippage. Two slippage scenarios are considered: (1) zero slippage, and (2) random slippage ranging from zero to two ticks, applied to both buy and close orders. This setup allows us to evaluate the robustness of the trading framework under realistic market conditions where execution prices may deviate slightly from expected levels.

4.1.3. Evaluation Metrics

To evaluate the classification performance of the trader selector, four widely used metrics were employed:

- Precision: Measures the proportion of correctly predicted positive instances among all instances predicted as positive. $\text{Precision} = \text{TP} / (\text{TP} + \text{FP})$
- Accuracy: Represents the ratio of correctly predicted instances to the total number of predictions, reflecting overall model correctness. $\text{Accuracy} = (\text{TP} + \text{TN}) / (\text{TP} + \text{TN} + \text{FP} + \text{FN})$
- Recall: Measures the ratio of true positive predictions to the total number of actual positive instances, indicating the model's ability to capture positive cases. $\text{Recall} = \text{TP} / (\text{TP} + \text{FN})$
- F1 Score: The harmonic mean of precision and recall, used to assess the balance between the two in classification performance. $\text{F1} = (2 \times \text{Precision} \times \text{Recall}) / (\text{Precision} + \text{Recall})$

To evaluate the framework's profitability and risk characteristics, the following metrics were employed:

- Return on Investment (ROI): $ROI = (\text{final asset} - \text{total cost}) / \text{total cost}$, where the total cost represents the cumulative cost over the entire evaluation period.
- Internal Rate of Return (IRR): Measures the annualized rate of growth of the investment, taking into account the timing and magnitude of cash flows. IRR is defined as the discount rate that sets the net present value (NPV) of all cash flows to zero:

$$0 = \sum_{t=0}^T \frac{C_t}{(1 + IRR)^t}$$

where C_t is the net cash flow at time t , calculated as the monthly invested capital plus or minus realized profits/losses. In this study, a fixed capital of TWD 500,000 is invested each month. If the previous month did not incur a loss, no additional capital injection is needed. This rule ensures that IRR reflects both the effect of periodic investments and the loss-compensation mechanism, providing an annualized measure of the strategy's growth under realistic trading conditions.

- Sharpe Ratio: Evaluates risk-adjusted performance by comparing excess returns to the standard deviation of returns, capturing the trade-off between return and volatility.

These experimental settings are aligned with those commonly adopted in prior futures trading studies, enabling fair comparison with existing high-return results.

4.1.4. Baselines and Comparison Methods

The Dual Thrust strategy is a classical trading approach and is commonly used as a baseline in many studies (M. Xu et al., 2024; Liu et al., 2020). This strategy requires three parameters: n , $k1$ and $k2$, where n represents the number of previous days considered, and $k1$ and $k2$ are adjustment coefficients. In this study, these parameters are set to $n = 5$, $k1 = 0.4$, and $k2 = 0.4$. In addition to the Dual Thrust strategy, the performance of other benchmark methods is evaluated for comparison:

- Buy & Hold (long): purchasing the asset at the beginning of the period and holding it without active trading.
- Short & Hold: taking a short position at the beginning of the period and maintaining it throughout.
- Accumulating ETF: We use 00663L.TW and 00675L.TW as our benchmarks. These two ETFs track the TAIEX and implement their strategies through futures trading.

These strategies serve as reference points to assess the effectiveness of the proposed trading approach across different market conditions, capturing both active trading performance and passive market exposure.

4.2. Results and Discussion

The performance of the proposed framework is evaluated using cumulative return, annualized return, and Sharpe ratio, and compared against the MTX index and benchmark ETFs.

4.2.1. Results of Individual Agents

As shown in Table 2, the short agent exhibits suboptimal performance, failing to generate any profit over the four-and-a-half-year backtesting period. In contrast, the long agent achieves a 458% ROI and a 53% IRR; however, its Sharpe Ratio is only 0.27, indicating relatively high risk-adjusted volatility. Overall, the performance of the long agent remains inferior to the baseline strategies, including Buy & Hold and Dual Thrust.

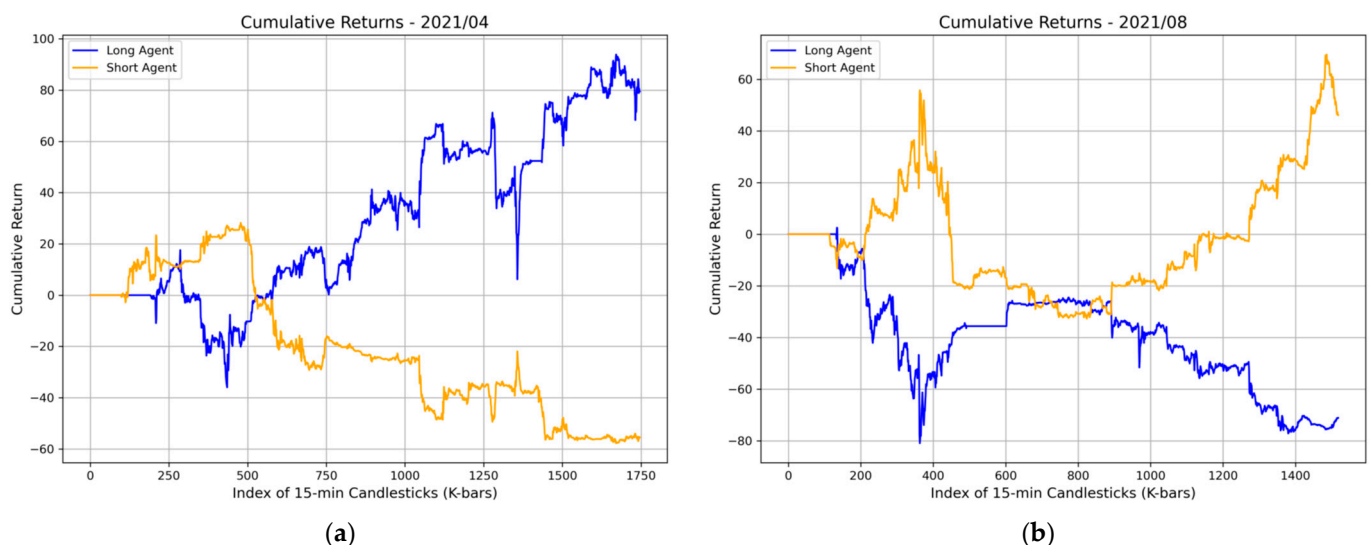
Table 2. Comparison of Two Agents and Baseline Strategy.

Trader	ROI (%)	IRR (%)	Sharpe Ratio
Long	458.00	53.00	0.29
Short	−89.00	−42.00	−0.32
Buy and Hold	845.00	46.20	0.52
Sell and Hold	−75.00	−29.00	−0.28
Dual Thrust	100.00	18.00	0.23
00675L.TW	569.60	39.60	N/A
00663L.TW	581.70	39.80	N/A

Note: ROI and IRR are reported in percentage terms. Sharpe ratio is calculated based on annualized returns and volatility. “N/A” indicates that the Sharpe ratio is not available due to data limitations.

Although the individual agents do not perform well overall, a more detailed analysis reveals that they can generate profits when operating in their respective areas of expertise. By examining the monthly performance of both agents, it is evident that they achieve profitability under favorable market conditions.

For example, Figure 6a illustrates the cumulative return during April 2021, with the horizontal axis representing the k-bar index and the vertical axis showing cumulative return. Correspondingly, Figure 7a presents the price movements for the same period, where the x-axis denotes the k-bar index and the y-axis denotes price level. From these figures, it can be observed that the overall market trend from the 0th to the 500th candlestick is downward. During this phase, the short agent is profitable. After the 500th candlestick, the market trend shifts upward, and the long agent’s profitability increases accordingly.

**Figure 6.** Cumulative Returns of Long and short Agents for April (a) and August (b) 2021.

Another example is shown in Figures 6b and 7b, which depict cumulative returns and price trends for August 2021. The price chart indicates an initial decline, followed by a rise, and then another decline. Figure 6b demonstrates that the short agent performs well during the downtrend, whereas the long agent exhibits strong performance during the uptrend.

These observations suggest that each agent can achieve strong performance when deployed under its respective favorable market conditions.

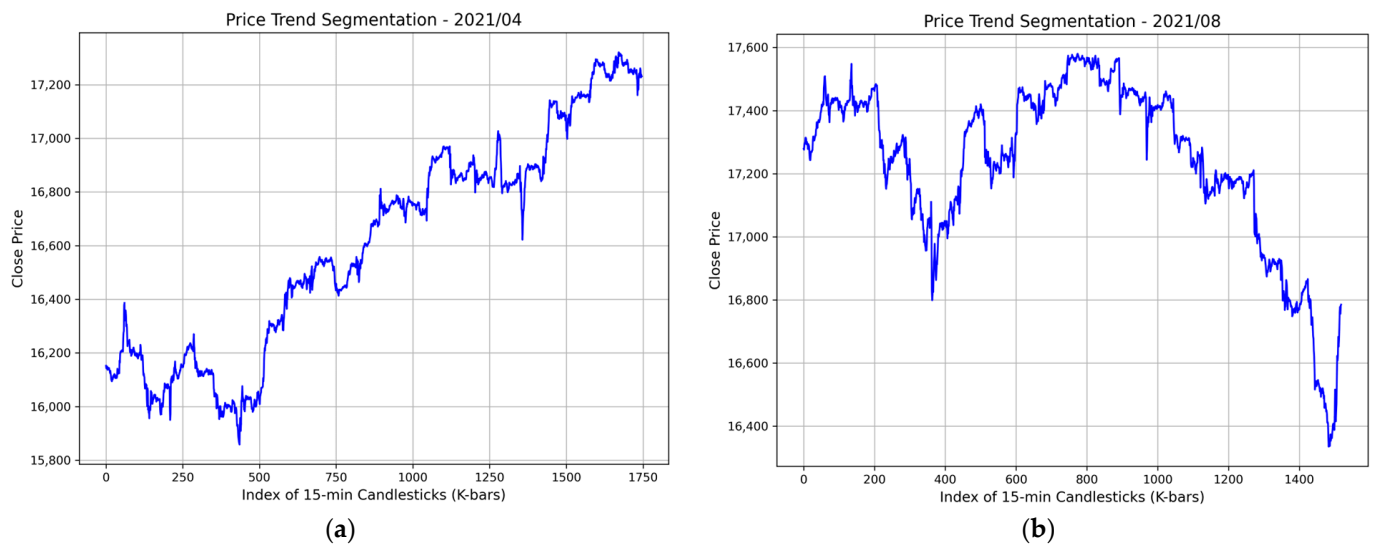


Figure 7. Price Trends for (a) April and (b) August 2021.

4.2.2. Framework Performance Evaluation via Simulation

This section presents a theoretical analysis based on simulations to investigate the impact of varying trader selector accuracies on the effectiveness of the trading strategy. The detailed trading procedure is described in Section 4.1.1. Historical MTX data from 2021 to 2025 were used for the simulations are summarized in Table 3.

Table 3. Performance at Different Accuracy Levels of the trader.

Accuracy (%)	ROI (%)	IRR (%)	Sharpe Ratio
50	41.46	−1.50	−0.04
60	1111.72	81.10	0.28
70	3477.17	141.80	0.35
80	6762.11	186.00	0.36
90	10,726.46	222.50	0.27
100	16,010.76	256.00	0.31

Note: Accuracy represents the prediction accuracy of the trader model. ROI and IRR are reported in percentage terms. Sharpe ratio is calculated using annualized return and annualized volatility.

The simulation results indicate that when the trader selector achieves a prediction accuracy of 60%, the cumulative return over the four-year period reaches 1112%, implying that the initial capital could grow more than elevenfold. The internal rate of return (IRR) also reaches 81%, reflecting strong profitability. However, the corresponding Sharpe Ratio is only 0.28, suggesting high return volatility. According to the Sharpe Ratio formula, which divides excess return by annualized volatility, a low Sharpe Ratio alongside a high IRR indicates that while the strategy generates substantial profits, it does so with relatively low stability.

These findings suggest that achieving a trader selector with approximately 60% or higher prediction accuracy allows the proposed trading framework to exhibit significant profit potential.

4.3. Practical Performance of the Proposed Framework

The training procedure described in Section 3.3 was used to train the trader selector, which was subsequently integrated into the proposed trading framework.

4.3.1. Trader Selector Evaluation

Table 4 presents the evaluation results of the trader selector across different half-year testing periods. The results show that the trader selector maintains solid classification performance in each period, with accuracy consistently around 70%, and F1 score, recall, and precision all near 0.7. These findings indicate that the model provides consistent and reliable predictive performance overall. Furthermore, when combined with the trading agents, the framework is able to generate profits in each half-year period.

Table 4. Performance metrics of the trader selector and profits for individual half-year periods without slippage price.

Selector	Accuracy (%)	F1 Score	Precision	Recall	Profit Without Slippage Price Setting (K, NTD\$)	Profit with Slippage Price Setting (K, NTD\$)
2021 H1	73.75	0.784	0.731	0.844	2510	2410
2021 H2	65.66	0.738	0.584	1.000	1254	1154
2022 H1	65.26	0.613	0.582	0.646	1215	1225
2022 H2	66.66	0.515	0.531	0.500	1410	1513
2023 H1	68.75	0.704	0.673	0.737	1619	961
2023 H2	66.66	0.515	0.531	0.500	1197	809
2024 H1	68.75	0.704	0.673	0.737	1490	1093
2024 H2	57.01	0.673	0.507	1.000	510	648
2025 H1	55.56	0.560	0.583	0.538	395	479
Average	65.73	0.645	0.600	0.722	1300	1144

Note: Accuracy is reported in percentage terms. F1 Score, Precision, and Recall are unitless metrics ranging from 0 to 1. Profit is reported in thousands of New Taiwan Dollars (K NTD\$). “Without slippage” refers to profit calculated ignoring transaction slippage, while “with slippage” includes slippage effects.

4.3.2. Framework Evaluation

The proposed framework was evaluated in the MTX futures market over the period from 2021 to 2025 and compared against multiple baseline strategies, including Buy and Hold, Sell and Hold, Dual Thrust, and leveraged ETFs such as 00663L.TW and 00675L.TW. The comparison results, summarized in Table 5, clearly demonstrate the superior performance of the proposed framework in terms of both ROI and IRR. Specifically, without considering market frictions such as slippage, the framework achieves an ROI of 2240% and an IRR of 109%, which represents a more than threefold improvement over the best-performing baseline, Buy and Hold (ROI 845%, IRR 46.2%). Even when realistic slippage effects are incorporated, the framework maintains a substantial advantage, with an ROI of 2058% and an IRR of 71%, highlighting its robustness under more practical trading conditions.

In addition to the high returns, the risk-adjusted performance of the framework is also notable. The Sharpe Ratio ranges from 0.31 without slippage to 0.35 with slippage, exceeding most baseline strategies. While these values indicate moderate risk-adjusted performance rather than exceptionally low volatility, they suggest that the framework is capable of generating substantial returns despite the inherent market fluctuations and leverage effects in futures trading. The decrease in IRR and the moderate Sharpe Ratio under the slippage setting further illustrate the importance of modeling realistic market frictions, as slippage naturally reduces net profitability and slightly increases exposure to short-term volatility.

Table 5. Comparison proposed framework and baseline model.

Method	ROI (%)	IRR (%)	Sharpe Ratio
Buy and Hold	845.00	46.20	0.52
Sell and Hold	−75.00	−29.00	−0.28
Dual Thrust	100.00	18.00	0.23
00663L.TW	569.60	39.60	N/A
00675L.TW	581.70	39.80	N/A
Proposed Framework without slippage setting	2240.00	109.00	0.31
Proposed Framework with slippage setting	2058.00	71.00	0.35

Note: ROI and IRR are reported in percentage terms. Sharpe ratio is calculated using annualized returns and volatility. “N/A” indicates that the Sharpe ratio cannot be calculated due to missing historical data. “Without slippage” and “with slippage” refer to profit calculation ignoring or including transaction slippage, respectively.

4.3.3. Trading Behavior Visualization

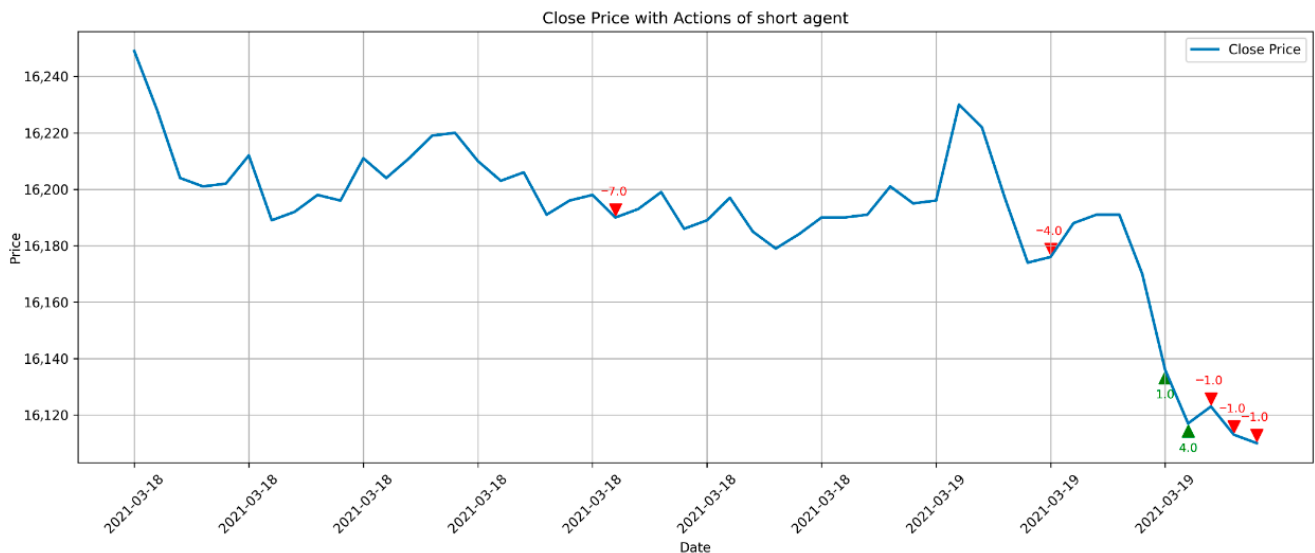
To provide a clearer understanding of the proposed trading framework’s behavior under real market conditions, this section presents its trading decisions over selected time periods.

Figure 8 illustrates the framework’s activity using the long agent during a specific period. The vertical axis shows the closing price, while the horizontal axis represents time. Triangles indicate executed trading actions at each time step, with positive values for buy actions and negative values for sell actions. Blue boxes denote the number of contracts held following each trade. During this period, the framework operated exclusively with the long agent.

**Figure 8.** Trading behavior of the proposed framework using the long agent during a selected period.

As shown in the figure, the agent typically opens long positions at relatively low prices and closes them at higher prices, illustrating its ability to capture profit opportunities in upward-trending markets.

Similarly, Figure 9 depicts the trading behavior of the framework when operating with the short agent during another selected period. The annotations follow the same format as in the previous figure, with triangles indicating trading actions. As illustrated, the agent generally opens short positions at relatively high prices and subsequently covers them at lower prices, highlighting its ability to capture profits in downward-trending markets.



may arise due to differences in market instruments, evaluation horizons, and capital management mechanisms, which further influence the balance between return and risk.

5. Conclusions

In this study, a novel trading framework is proposed, consisting of two agents and a trader selector designed to dynamically choose the most suitable agent based on daily market trends. The framework is evaluated through backtesting in the MTX market, where transaction fees and margin requirements are incorporated to ensure a realistic trading environment.

Extensive empirical experiments are conducted to compare the proposed framework with the classical Dual Thrust strategy. The results demonstrate that the proposed approach significantly outperforms the traditional strategy in the MTX market. A four-year historical backtest is performed, covering critical events such as the tax adjustment and the Russia–Ukraine war. Even under these extreme market conditions, the framework consistently achieves stable and considerable profits, demonstrating both resilience and practical applicability.

Our framework achieves an ROI of 2240% without slippage and 2058% with slippage, along with an IRR of 109% and 73%, respectively. Compared to the benchmark strategies presented in this paper, the framework attains at least twice the ROI of the best-performing baselines. Furthermore, simulations of the trader selector under varying prediction accuracies indicate that once accuracy reaches 60% or higher, the framework consistently delivers stable and satisfactory performance. Backtesting results further confirm that the proposed method significantly outperforms all baseline strategies in terms of both ROI and IRR.

To further enhance the robustness and practical relevance of the proposed framework, several future research directions are identified. First, improving the trader selection mechanism with more advanced decision models is a promising avenue, as overall profitability is closely tied to the accuracy of agent selection. Second, the realism of the trading environment can be strengthened by incorporating additional market frictions, such as slippage and liquidity constraints, which are prevalent in real-world trading and would allow simulations to more accurately reflect actual market conditions. Third, integrating more comprehensive risk management mechanisms—including stop-loss rules and dynamic position sizing—could further improve performance stability and resilience across different market regimes. Finally, future experiments may adopt a lower reset frequency (i.e., longer reset intervals) to provide a more stringent and realistic evaluation of long-term trading performance. In addition, exploring how to deploy the framework in real-world trading environments represents an important direction for future work.

Author Contributions: Conceptualization, Y.-H.H. and C.-H.L.; methodology, Y.-H.H.; software, Y.-H.H. and C.-H.L.; validation, Y.-H.H., C.-H.L. and S.-M.Y.; formal analysis, Y.-H.H.; investigation, Y.-H.H., C.-H.L. and S.-M.Y.; resources, Y.-H.H. and C.-H.L.; data curation, Y.-H.H.; writing—original draft preparation, C.-H.L.; writing—review and editing, Y.-H.H. and C.-H.L.; visualization, Y.-H.H. and C.-H.L.; supervision, S.-M.Y.; project administration, S.-M.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The quantitative data were obtained from SinoPac Securities. The source code is available on GitHub at: https://github.com/nctu-dcs-lab/Trading_Framework (accessed on 15 January 2026).

Conflicts of Interest: The authors declare no conflicts of interest.

References

- Al-Khasawneh, M. A., Raza, A., Khan, S. U. R., & Khan, Z. (2025). Stock market trend prediction using deep learning approach. *Computational Economics*, 66(1), 453–484. [\[CrossRef\]](#)
- Beck, M., Pöppel, K., Spanring, M., Auer, A., Prudnikova, O., Kopp, M., Klambauer, G., Brandstetter, J., & Hochreiter, S. (2024). xlstm: Extended long short-term memory. *Advances in Neural Information Processing Systems*, 37, 107547–107603.
- Biswas, A. K., Bhuiyan, M. S. A., Mir, M. N. H., Rahman, A., Mridha, M. F., Islam, M. R., & Watanobe, Y. (2025). A dual output temporal convolutional network with attention architecture for stock price prediction and risk assessment. *IEEE Access*, 13, 53621–53639. [\[CrossRef\]](#)
- Challu, C., Olivares, K. G., Oreshkin, B. N., Ramirez, F. G., Canseco, M. M., & Dubrawski, A. (2023). Nhits: Neural hierarchical interpolation for time series forecasting. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37, 6989–6997. [\[CrossRef\]](#)
- Chen, D.-Y., Wu, C.-W., & Wang, C.-W. (2025). Using CNN models to predict the future trends of listed stocks on the Taiwan stock exchange. In *International conference on human-computer interaction*. Springer Nature.
- Das, A., Kong, W., Leach, A., Mathur, S., Sen, R., & Yu, R. (2023). Long-term forecasting with tide: Time-series dense encoder. *arXiv*, arXiv:2304.08424.
- Ge, Q. (2025). Enhancing stock market Forecasting: A hybrid model for accurate prediction of S&P 500 and CSI 300 future prices. *Expert Systems with Applications*, 260, 125380.
- Gil, G. L., Duhamel-Seblin, P., & McCarren, A. (2024). An evaluation of deep learning models for stock market trend prediction. *arXiv*, arXiv:2408.12408. [\[CrossRef\]](#)
- Heess, N., Hunt, J. J., Lillicrap, T. P., & Silver, D. (2015). Memory-based control with recurrent neural networks. *arXiv*, arXiv:1512.04455. [\[CrossRef\]](#)
- Hochreiter, S., & Schmidhuber, J. (1996, December 2–5). *LSTM can solve hard long time lag problems*. Advances in Neural Information Processing Systems, 9, Denver, CO, USA.
- Huang, S., & Ontañón, S. (2020). A closer look at invalid action masking in policy gradient algorithms. *arXiv*, arXiv:2006.14171. [\[CrossRef\]](#)
- Kabbani, T., & Duman, E. (2022). Deep reinforcement learning approach for trading automation in the stock market. *IEEE Access*, 10, 93564–93574. [\[CrossRef\]](#)
- Kirtac, K., & Germano, G. (2024). Sentiment trading with large language models. *Finance Research Letters*, 62, 105227. [\[CrossRef\]](#)
- Lea, C., Flynn, M. D., Vidal, R., Reiter, A., & Hager, G. D. (2017, July 21–26). *Temporal convolutional networks for action segmentation and detection*. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA.
- Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., & Wierstra, D. (2015). Continuous control with deep reinforcement learning. *arXiv*, arXiv:1509.02971.
- Lim, B., Arık, S. Ö., Loeff, N., & Pfister, T. (2021). Temporal fusion transformers for interpretable multi-horizon time series forecasting. *International Journal of Forecasting*, 37(4), 1748–1764. [\[CrossRef\]](#)
- Liu, Y., Liu, Q., Zhao, H., Pan, Z., & Liu, C. (2020, February 7–12). *Adaptive quantitative trading: An imitative deep reinforcement learning approach*. Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA.
- Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., & Ostrovski, G. (2015). Human-level control through deep reinforcement learning. *Nature*, 518(7540), 529–533. [\[CrossRef\]](#) [\[PubMed\]](#)
- Oreshkin, B. N., Carpov, D., Chapados, N., & Bengio, Y. (2019). N-BEATS: Neural basis expansion analysis for interpretable time series forecasting. *arXiv*, arXiv:1905.10437.
- Padmanayana, V., & Bhavya, K. (2021). Stock market prediction using Twitter sentiment analysis. *International Journal of Scientific Research in Science and Technology*, 7(4), 265–270. [\[CrossRef\]](#)
- Peng, Y.-L. (2025, June 25–27). *Probability-driven hedge ratio optimization using machine learning: An adaptive approach to financial risk management*. 2025 IEEE/ACIS 29th International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD), Busan, Republic of Korea.
- Qin, M., Cai, X., Li, Y., Xia, H., Zong, C., Sun, S., Wang, X., & An, B. (2025). FineFT: Efficient and risk-aware ensemble reinforcement learning for futures trading. *arXiv*, arXiv:2512.23773.
- Schulman, J., Levine, S., Abbeel, P., Jordan, M., & Moritz, P. (2015a, July 6–11). *Trust region policy optimization*. International Conference on Machine Learning, Lille, France.
- Schulman, J., Moritz, P., Levine, S., Jordan, M., & Abbeel, P. (2015b). High-dimensional continuous control using generalized advantage estimation. *arXiv*, arXiv:1506.02438.
- Schulman, J., Wolski, F., Dhariwal, P., Radford, A., & Klimov, O. (2017). Proximal policy optimization algorithms. *arXiv*, arXiv:1707.06347. [\[CrossRef\]](#)

- Schuster, M., & Paliwal, K. K. (1997). Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11), 2673–2681. [\[CrossRef\]](#)
- Sutton, R. S., & Barto, A. G. (1998). *Reinforcement learning: An introduction* (Vol. 1). MIT Press.
- Towers, M., Kwiatkowski, A., Terry, J., Balis, J. U., De Cola, G., Deleu, T., Goulão, M., Kallinteris, A., Krimmel, M., & KG, A. (2024). Gymnasium: A standard interface for reinforcement learning environments. *arXiv*, arXiv:2407.17032. [\[CrossRef\]](#)
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017, December 4–9). *Attention is all you need*. Advances in Neural Information Processing Systems 30, Long Beach, CA, USA.
- Wang, S. (2023). A stock price prediction method based on BiLSTM and improved transformer. *IEEE Access*, 11, 104211–104223. [\[CrossRef\]](#)
- Xu, M., Lan, Z., Tao, Z., Du, J., & Ye, Z. (2024, May 24–26). *Deep reinforcement learning for quantitative trading*. 2024 4th International Conference on Electronics, Circuits and Information Engineering (ECIE), Hangzhou, China.
- Xu, Y., & Cohen, S. B. (2018, July 15–20). *Stock movement prediction from tweets and historical prices*. Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Melbourne, Australia.
- Yang, H., Liu, X.-Y., Zhong, S., & Walid, A. (2020, October 15–16). *Deep reinforcement learning for automated stock trading: An ensemble strategy*. Proceedings of the First ACM International Conference on AI in Finance, Manhattan, NY, USA.
- Yu, X., Wu, W., Liao, X., & Han, Y. (2023). Dynamic stock-decision ensemble strategy based on deep reinforcement learning. *Applied Intelligence*, 53(2), 2452–2470. [\[CrossRef\]](#) [\[PubMed\]](#)
- Zhang, J., Ye, L., & Lai, Y. (2023). Stock price prediction using CNN-BiLSTM-Attention model. *Mathematics*, 11(9), 1985. [\[CrossRef\]](#)
- Zhang, Z., Zohren, S., & Roberts, S. (2019). Deep reinforcement learning for trading. *arXiv*, arXiv:1911.10107. [\[CrossRef\]](#)
- Zhao, Y., & Yang, G. (2023). Deep learning-based integrated framework for stock price movement prediction. *Applied Soft Computing*, 133, 109921. [\[CrossRef\]](#)

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.